

Technická univerzita v Liberci

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B 2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Elektronické řízení plynového pedálu pro tzv. pomalý rozjezd

Electronic control of throttle for partial load

Bakalářská práce

| | |
|----------------|---------------------|
| Autor: | Josef Bednář |
| Vedoucí práce: | Ing. Tomáš Pluhař |
| Konzultant: | Ing. Pavel Pírk |

V Liberci 4. 4. 2008

!!! SEM vložit originál zadání práce ... !!!

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé BP a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím práce a konzultantem.

Datum:

Podpis:

Abstrakt

Cílem práce je navrhnout a vytvořit prototyp zařízení, které bude sloužit k simulaci sešlápnutí plynového pedálu ve vozech Škoda. Zařízení bude využíváno pro účely laboratorních akustických testů na automobilech, konkrétně pro měření vnitřního hluku při pomalém rozjezdu. Pomalý rozjezd je definován tak, že v průběhu rozjezdu z volnoběžných otáček motoru po maximální otáčky příslušné motorizace nesmí zrychlení přesáhnout hodnotu $0,5 \text{ m/s}^2$. Tuto definici je schopen ještě dodržet člověk s určitou zkušeností, ale při řízení člověkem není možné nikdy dva následující testy považovat za identické. Požadavek na opakovatelnost měření je jeden z hlavních důvodů, proč v tomto druhu testu použít elektronické zařízení, které bude schopné těmto podmínkám vyhovět.

Zařízení musí být univerzální pro všechny typové řady automobilů Škoda, a proto je zapotřebí otáčky respektive rychlost automobilu řídit ve zpětnovazební smyčce. Údaje o otáčkách motoru a rychlosti automobilu jsou k dispozici na diagnostické sběrnici, která používá protokol sběrnice CAN. Dále je v automobilu k dispozici konektor pro připojení plynového pedálu k řídící jednotce motoru.

Spolu s přístrojem je zapotřebí navrhnout ovládací aplikaci, ve které bude vedena databáze automobilů včetně jejich parametrů. Z této databáze si bude uživatel volit automobil k testu. Vozidla do databáze bude vkládat uživatel prostřednictvím jednoduchého prostředí a existující databázi bude možno v tomto prostředí editovat. Komunikace mezi samotným zařízením a aplikací na PC bude realizována po sběrnici USB, aby bylo možné zařízení snadno připojit k většině počítačů včetně notebooků.

Klíčová slova: Mikroprocesor, zpětnovazební řízení, CAN, USB

Abstract

The purpose of this work is to design and create prototype of a device, which will simulate a throttle pedal in Skoda cars. This device will be used for laboratory acoustic tests on cars, exactly for so-called partial load test. The car is accelerating from idle speed to maximal speed of current engine during this test and the acceleration must not exceed $0,5 \text{ m/s}^2$. Skilled person can sometimes fulfill this requirement, but the repeatability of the measurement is the main reason for using of electronic device to control the car. The equipment has to be universal for all types of Skoda cars. Therefore it is essential to control engine speed and car velocity in a feedback loop. There are useful data available on the car diagnostic bus. The bus use CAN protocol. There is a throttle connector also available in the car.

The device setup should be performed by the computer application which is essential to developed for this purpose. The database will be created in the application. There will be important parameters of cars stored in the database. Operator will choose a car for test from this database. Operator can also easily add another car or edit the existing one using simple forms of the application. Communication between the application and the device will be realized via USB to provide easy connectivity to most nowadays computers including notebooks.

Keywords: Microcontroller, Feedback loop driving, CAN, USB

Obsah

| | |
|---|-----------|
| ABSTRAKT..... | 4 |
| ABSTRACT | 5 |
| OBSAH..... | 6 |
| SEZNAM ZKRATEK..... | 8 |
| SEZNAM SYMBOLŮ | 8 |
| ÚVOD..... | 9 |
| 1 HARDWARE | 10 |
| 1.1 Teoretický rozbor | 10 |
| 1.2 Mikroprocesor..... | 13 |
| 1.2.1 Oscilátor..... | 14 |
| 1.2.2 A/D převodník | 16 |
| 1.2.3 D/A převodník | 17 |
| 1.2.4 Sériové rozhraní UART | 17 |
| 1.2.5 Časovače | 18 |
| 1.2.6 Řadič sběrnice CAN | 20 |
| 1.2.7 Implementace zpětnovazební řídicí smyčky | 24 |
| 1.2.8 Testovací programy | 30 |
| 1.2.9 Finální verze programu pro procesor | 34 |
| 1.3 FTDI Chip | 38 |
| 1.3.1 Popis použitého obvodu | 38 |
| 1.3.2 Konkrétní zapojení obvodu..... | 38 |
| 1.3.3 Programování FT232RL | 39 |
| 1.4 Budič sběrnice CAN | 40 |
| 1.5 Koncový stupeň pro připojení k plynovému pedálu | 40 |
| 1.6 Napájecí modul | 42 |

| | | |
|----------|---|-----------|
| 1.7 | Plošné spoje | 44 |
| 2 | OVLÁDACÍ APLIKACE..... | 47 |
| 2.1 | Základní princip a požadavky | 47 |
| 2.2 | Pracovní prostředí | 48 |
| 2.3 | Komunikační protokol | 50 |
| 3 | MANUÁL PŘÍSTROJE | 52 |
| 3.1 | Úvod | 53 |
| 3.2 | Přístroj Gas control | 53 |
| 3.2.1 | Rozložení prvků na přístroji..... | 54 |
| 3.2.2 | Kabely pro připojení k pedálu..... | 55 |
| 3.2.3 | Režimy přístroje..... | 56 |
| 3.2.4 | Popis ovládacích prvků | 57 |
| 3.2.5 | Popis signalizačních prvků..... | 57 |
| 3.3 | Počítačová aplikace Gas control..... | 58 |
| 3.3.1 | Popis pracovního prostředí..... | 58 |
| 3.4 | Pokyny a doporučené postupy | 61 |
| 3.4.1 | Instalace aplikace | 61 |
| 3.4.2 | Používání aplikace | 62 |
| 3.4.3 | Používání sady Gas control..... | 63 |
| 3.4.4 | Kalibrace koncového stupně pro pedál | 64 |
| 4 | ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ | 66 |
| 5 | ZÁVĚR | 67 |

Seznam zkratek

CAN bus – Controller Area Network – sériová diferenciální sběrnice

A/D převodník – Analogovo-digitální převodník

D/A převodník – Digitálně-analogový převodník

SAR – Successive Aproximation Register – A/D převodník s postupnou aproximací

I/O porty – Vstupně výstupní porty

Paměť RAM – Random Access Memory – Volatilní paměť procesoru

Paměť FLASH – Nevolatilní paměť mikroprocesoru

UART – Universal Assynchronous Receiver Transmitter

USB – Universal Serial Bus

JTAG – Joint Test Action Group

SYSCCLK – Systémový hodinový signál mikroprocesoru

SFR – Special Function Registers – registry speciálních funkcí procesoru

SMD – Surface Mounted Device – součástky pro povrchovou montáž

API – Application Programming Interface – funkce pro programování pod OS windows

IDE – Integrated Development Environment – vývojové prostředí

Uchar – Unsigned char – Datový typ jazyka C

Seznam symbolů

MIPS – milion instrukcí za sekundu

b – bit – jednotka informace

B – bajt – jednotka informace

Úvod

Již ze zadání vyplývá, že celá práce bude rozdělena na dvě části. Tato základní struktura bude dodržována v celé práci kromě výjimek, které nutně spojují obě části.

První a rozsáhlejší část bude pojednávat o hardwaru zařízení. Do této části bude zahrnuto vše, co se týká mikroprocesoru, jako hlavní výpočetní součástky, jeho využívaných periferií a externích obvodů potřebných pro funkci přístroje. Dále zde bude probráno vše okolo návrhu vlastního zařízení, což je návrh napájení obvodů, návrh desek plošných spojů a návrh konečného fyzického provedení přístroje.

Druhá část práce se bude týkat vývoje počítačové aplikace, která bude sloužit k ovládání přístroje. Bude zde popsáno pracovní prostředí aplikace, základy komunikace po sériové lince s přístrojem a také návrh komunikačního protokolu, který je samozřejmě implementovaný v obou částech, ale datová struktura vzniká v aplikaci na PC.

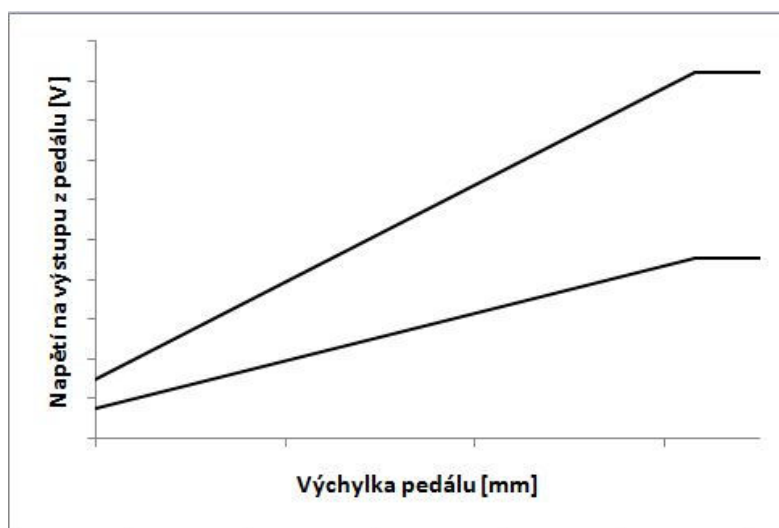
Celá práce byla zadána z důvodů potřeby regulovat co nejpřesněji rychlost automobilu při pomalém rozjezdu. V současnosti není na trhu podobné zařízení. Jediný přístroj, který alespoň přibližně slouží k těmto účelům, je cenově velice náročné zařízení, které pro řízení používá mechanické stlačování plynového pedálu („Robot driver“ od firmy Froude Hofmann).

1 Hardware

1.1 Teoretický rozbor

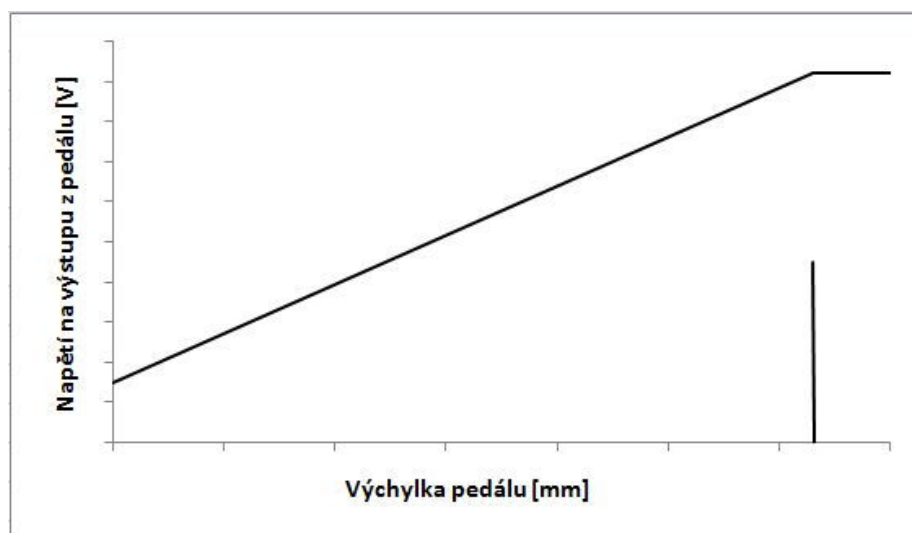
Základní funkce přístroje je, aby byl schopný fungovat jako reálný plynový pedál v automobilu. Tímto pro automobil přirozeným způsobem bude přístroj ovládat motor automobilu. Plynové pedály v dnešní generaci automobilů fungují na základě elektronických signálů, které jsou vysílány do řídicí jednotky motoru. Při realizaci přístroje tedy nebude nutné používat elektro-mechanické převodníky. U automobilů, které přístroj musí umět obsluhovat, se vyskytují různé varianty provedení plynových pedálů. Starší řady automobilů snímají polohu pedálu pomocí potenciometrů a novější řady pomocí hallových snímačů. Všechny pedály mají vlastní vyhodnocovací obvod, takže na výstupu z pedálu jsou už jen stejnosměrné napěťové úrovně. Jejich velikost je úměrná poloze pedálu. Starší provedení pedálu s potenciometry se dále ještě dělí do dvou provedení, které se navzájem liší počtem výstupních napětí. Jedna varianta je použita v automobilech se zážehovým motorem a druhá u automobilů s motorem vznětovým.

U zážehových motorů má plynový pedál zapojení se dvěma potenciometry. Potenciometr je zde zapojen jako napěťový dělič, takže na výstupu plynového pedálu jsou dvě napěťové úrovně, od každého potenciometru jedna. Napětí na výstupu pedálu jsou lineárními funkcemi výchylky plynového pedálu. Směrnice jednoho napětí je v této variantě dvojnásobná oproti směrnici druhého, jak je zobrazeno na obr. 1.



Obr. 1: Závislost výstupních signálů z plynového pedálu na jeho výchylce

V ideálním případě má jedno napětí dvojnásobnou hodnotu oproti druhému. Řídící jednotka zpracovává oba signály zároveň a ze souběhu obou signálů vyhodnocuje případné poruchy pedálu. Pokud se jedno z napětí vyskytne mimo tolerance, nahlásí řídící jednotka chybu. Tato chyba však není fatální, protože motor pracuje dál na základě obou napětí. Horší situace nastane, pokud je na výstupu pedálu jen jedno napětí nebo dokonce žádné. K takovým situacím nesmí docházet, protože řídící jednotka začne motor řídit sama a nastaví ho do zvýšených volnoběžných otáček. Tato funkce slouží pro dojetí v případě poruchy pedálu. Takovýmito situacím je potřeba předejít tak, že před nastartováním automobilu musí být přístroj připojený a zapnutý, aby na vstupu řídící jednotky již byla obě napětí. U druhé varianty použité u vznětových motorů je v zapojení pouze jeden potenciometr a jeden přepínač. Na výstupu je jen jedna napěťová úroveň a diskrétní signál, signalizující maximální sešlápnutí pedálu, jak je zobrazeno na obr. 2.



Obr. 2: Závislost výstupních signálů z plynového pedálu na jeho výchylce

Chyba způsobená nesouběžným napětím u této varianty odpadá, ale k chybě způsobené chybějícím napětím docházet může, takže zde platí stejná opatření jako u zážehových motorů. U novějších verzí s hallovými indukčními snímači jsou na výstupu z pedálu dvě napěťové úrovně stejně jako u starší verze pro zážehové motory (obr. 1).

Z těchto specifikací je jasné, že základním požadavkem na procesor, který bude řídit celý proces, budou jeho A/D a D/A převodníky. Základní úvahou, jak simulaci pedálu realizovat, je nastavovat napětí podle daných pravidel pomocí D/A převodníku s poměrně malou periodou. Signál sice nebude spojitý jako v případě reálného pedálu, ale při periodě v řádu milisekund lze dosáhnout poměrně uspokojivého signálu. Původní

záměr byl takový, že u pedálu se dvěma napětími bude každý ze dvou D/A převodníků generovat jeden signál. V průběhu testování ale docházelo neustále k chybám způsobeným špatným souběhem obou signálů. Příčinou bylo pravděpodobně nastavování dvou převodníků, které není z praktického pohledu možné vyvolat v jeden okamžik. Řídicí jednotka tento malý časový posun mezi signály vyhodnocovala už jako chybu na pedálu. Výstupní signál z D/A převodníku musí být navíc ještě napětově zesílen, aby jeho hodnota mohla být vyšší než 3 V, což je maximum napětí na procesoru. Proto byl mezi procesor a řídicí jednotku zařazen koncový stupeň, který zařídí jednak zesílení signálu na požadovanou úroveň v automobilu a také přesný souběh napětí. Pro přesné dodržení souběhu bude koncový stupeň vybaven kalibračním odporovým trimrem.

V klidovém režimu se přístroj bude chovat transparentně, takže by uživatel neměl poznat, že je mezi plynový pedál a řídicí jednotku přístroj zapojen. V tomto režimu bude napětí na reálném pedálu zjišťováno pomocí A/D převodníku a stejné napětí nastavováno opět pomocí D/A převodníku jako v případě automatického režimu. Realizace klidového režimu by šla vyřešit i jinými způsoby, např. pomocí bezkontaktního přepínání signálu z pedálu nebo procesoru, ale při přepínání by mohlo docházet k hazardním stavům a řídicí jednotka by je mohla vyhodnotit jako selhání pedálu a motor uvést do nouzového režimu, jak je popsáno výše. Ve většině dnešních procesorů jsou navíc A/D i D/A převodníky implementovány běžně, takže s těmito požadavky při výběru procesoru problémy nebudou.

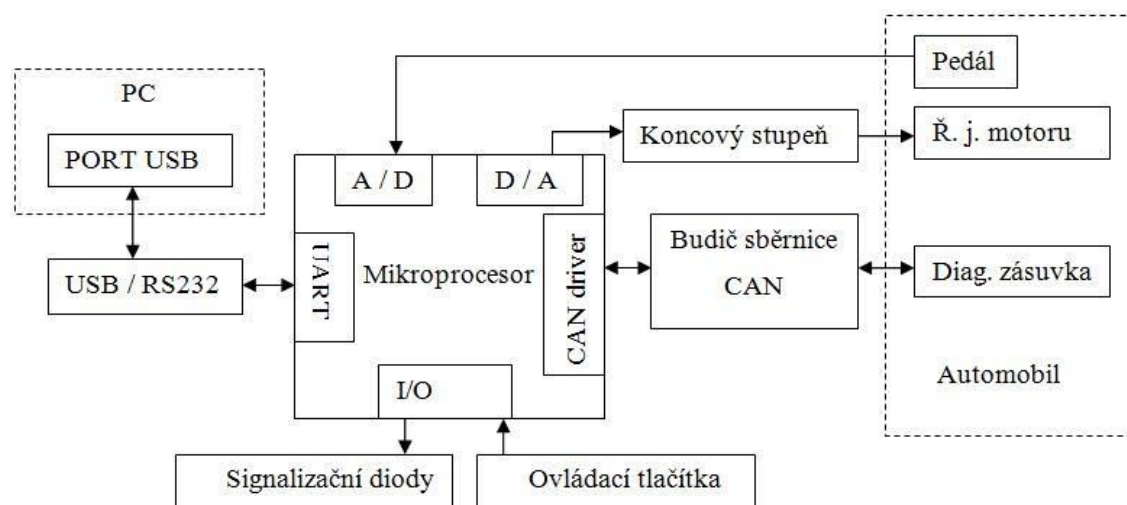
Neméně důležitým požadavkem na mikroprocesor je integrovaný řadič sběrnice CAN. Tento požadavek není sice absolutně nutný, protože CAN řadiče se dají pořídit i jako samostatné obvody, které pak komunikují s nadřazeným mikroprocesorem po sériové nebo paralelní sběrnici, ale pokud nic nebrání tomu použít procesor s integrovaným řadičem, tak pro zjednodušení celého zapojení je lepší použít řadič integrovaný.

V ideálním případě by procesor měl mít také integrované rozhraní sběrnice USB, která bude využívána pro přenos dat z ovládací aplikace do procesoru. V běžném sortimentu současných procesorů se bohužel nevyskytuje žádný klon, který by měl integrované rozhraní USB a zároveň řadič CAN. Pro konkrétní aplikaci bude tedy jednodušší použít procesor s integrovaným CAN řadičem. Většina dnešních procesorů má totiž integrovaný UART, který lze snadno použít pro komunikaci po sériové lince. Protože zařízení bude používáno hlavně s notebooky, ze kterých v poslední době

sériový port mizí, přístroj bude využívat externí integrovaný obvod pro konverzi mezi rozhraními USB a obecnou sběrnici UART.

Ostatní běžné požadavky na mikroprocesory jako např. výpočetní výkon, dostatek I/O portů a dostatečná paměť RAM a FLASH lze při výběru procesoru opomenout, protože v těchto ohledech aplikace na mikroprocesor nebude klást velké nároky. S ohledem na všechny zmíněné parametry byl nakonec vybrán procesor C8051F040 firmy SILABS (Silicon Laboratories).

Tímto je tedy nastíněna základní podoba hardwarové konfigurace, která je zobrazena v blokovém schématu celého zařízení (obr. 3). Hlavním prvkem je mikroprocesor s integrovaným řadičem CAN, ke kterému musí být připojený budič CAN sběrnice a další náležitosti pro připojení k diagnostické sběrnici. K procesoru bude připojen i stávající plynový pedál. Jeden napěťový signál bude zdrojem signálu pro A/D převodník. Signály z D/A převodníků budou výstupními signály z přístroje a po úpravě signálů koncovým stupněm i vstupním signálem do řídicí jednotky motoru. K procesoru bude připojen i obvod pro konverzi UART na USB a také ovládací tlačítka a signalizační diody.



Obr. 3: Blokové schéma hardwarové konfigurace přístroje

1.2 Mikroprocesor

Pro vyvíjenou aplikaci byl vybrán procesor C8051F040 od firmy Silicon laboratories. Tato firma vyrábí několik řad vyspělých mikroprocesorů s jádrem 8051 a širokým spektrem zabudovaných periférií. Řada F04x, ze které je používán procesor se vyznačuje následujícími parametry:

- Výpočetní výkon až 25 MIPS
- 32 / 64 digitálních I/O portů
- Integrovaný řadič CAN 2.0 B
- 12b A/D převodník s analogovým multiplexorem
- 12b D/A převodníky
- Hardwarově implementované sériové linky UART
- 5 časovačů (až 16b)
- 4352 B paměti RAM
- 64 kB programovatelné paměti FLASH
- JTAG systém umožňující ladění programu přímo v čipu
- Atd.¹

Pro vývoj zařízení byl mikroprocesor zakoupen ve vývojové sadě C8051F040-DK, kde je procesor již osazen spolu s nejn nutnějšími externími součástkami a většina využitelných periférií je vyvedena na piny desky. Vývojová sada se skládá z vývojové desky, napájecího zdroje, USB/JTAG programátoru a potřebného softwaru. Vývojová deska byla využita v celém procesu vývoje a odladění programu, hlavně u prvních testovacích programů. Vývoj kódu přístroje byl proveden ve vývojovém prostředí Silicon Laboratories IDE dodávaném spolu s vývojovou sadou. Jako kompilátor byl využíván program Keil (www.keil.com), který je v sadě dodáván jako omezená verze s maximálním počtem instrukcí 4kB.

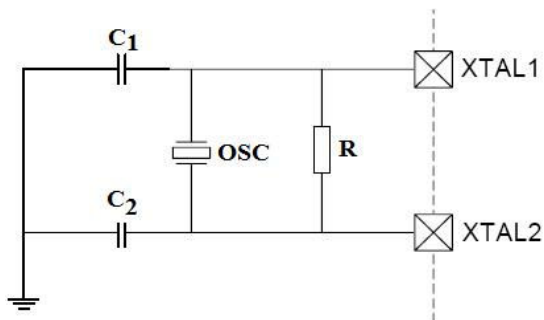
Při následujícím popisu periférií procesoru C8051F040 budou používány názvy registrů z oficiálního katalogového listu [1].

1.2.1 Oscilátor

Všechny procesory řady F04x mají integrovaný programovatelný oscilátor, který je vždy nastavený jako zdroj hodinového signálu po resetu. Tovární nastavení interního oscilátoru je na 24,5 MHz. Hodnota jeho periody je v malém rozsahu programovatelná pomocí registru OSCICL, ale frekvence oscilátoru nesmí překročit 25 MHz. Systémový hodinový signál (SYSCLK) může být odvozen od interního oscilátoru zařazením

¹ Uvedené parametry jsou pouze vybrané parametry, které jsou důležité pro zpracovávanou aplikaci.

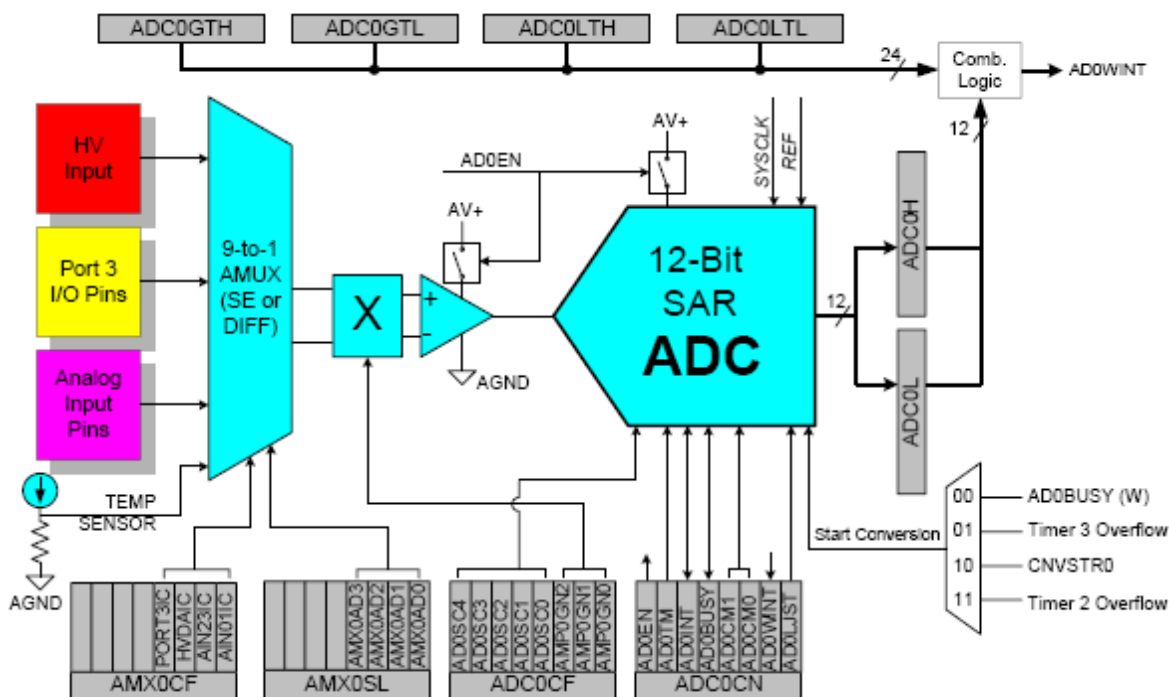
předděličky hodinového signálu, jejíž hodnota je nastavena v registru OSCICN. Pro většinu aplikací, které nejsou náročné na přesné časování rychlých dějů, interní oscilátor stačí, ovšem pro používání protokolu sběrnice CAN bus potřebuje procesor externí oscilátor s frekvencemi, které jsou doporučeny respektive předepsány ve specifikaci sběrnice [2]. Procesor je vybaven obvodem pro přímé připojení různých externích oscilátorů. Ty se pouze připojí k vývodům XTAL1 a XTAL2 s minimálním počtem externích součástek, jak je vidět na obr. 4.



Obr. 4: Schéma připojení externího oscilátoru k procesoru

V programu, kde bude využíván externí oscilátor, musí proběhnout v inicializačních funkcích nastavení procesoru pro používání externího oscilátoru, protože při spuštění resp. po resetu je jako výchozí nastaven interní oscilátor. Po nastavení všech parametrů v registru OSCXCN dojde k rozběhnutí oscilátoru. Pokud by se program v době, kdy ještě není oscilátor ustálen na pracovní frekvenci, přepnul do používání externího oscilátoru, může dojít k nepředvídatelnému chování procesoru. Procesor má hardwarové prostředky, kterými hlídá ustálení externího oscilátoru a po ustálení nastaví příznak v registru OSCXCN. Aby program předcházel neočekávaným selháním, musí být v programu tento příznak hlídán. Po ustálení může být procesor přepnut do režimu, ve kterém jako zdroj hodinového signálu používá externí oscilátor. Tato volba se provádí v registru CLKSEL.

1.2.2 A/D převodník



Obr. 5: Schéma A/D subsystému s ovládacími registry

Používaný procesor má k dispozici jeden osmibitový A/D převodník (ADC2) a jeden dvanáctibitový převodník (ADC0), který bude programem využíván. Subsystem A/D převodníku na procesoru se skládá z těchto částí: 9kanálový programovatelný multiplexor (AMUX0), programovatelný vstupní zesilovač (PGA0) a samotný 12b A/D převodník typu SAR (successive approximation register – převodník s postupnou aproximací). Tento převodník může pracovat se vzorkovací frekvencí až 100 000 vzorků/s. Schéma celého A/D subsystému je zobrazeno na obr. 5. Všechny jeho součásti jsou plně programovatelné přes SFR registry (Special Function Registers) procesoru (viz obr. 5). V programu bude využíván pouze jeden analogový vstup. Vstupní předzesilovač bude nastaven na hodnotu zesílení 0,5. Toto nastavení je nezbytné pro pokrytí celého rozsahu napětí na plynovém pedálu s interní referencí. Napětí na reálném pedálu je podle specifikace v rozsahu 0-5 V ovšem jeho reálné hodnoty nepřesahují 4,75 V, což bylo ověřeno měřením. Proto lze s interním referenčním napětím 2,4 V a předzesilovačem vzorkovat celý rozsah.

Při inicializaci A/D převodníku musí být nastaven zdroj referenčního napětí (registr REF0CN). Program bude využívat interní referenční napětí 2,4 V (viz výše). Dále musí být nastavena frekvence, respektive takt, konverze a zisk vstupního

zesilovače (registr ADC0CF), konfigurace analogových vstupů a multiplexoru (registry AMX0SL a AMX0CF) a zdroj pro spuštění konverze (registr AD0CN). Konverze může být spouštěna cyklicky při přetečení jednoho z časovačů nebo v ručním režimu, kdy konverze začne při programovém nastavení bitu AD0BUSY v registru AD0CN. Pokud není využíváno přerušení od A/D převodníku, které signalizuje ukončení konverze, tak před přečtením hodnoty z registrů ADC0H a ADC0L, musí být v programu sledováno nastavení příznaku AD0INT (příznak dokončené konverze) a teprve, když dojde k jeho hardwarovému nastavení, je hodnota ve výstupních registrech připravena k použití. Ve výstupních registrech, které mají dohromady velikost 16 b, může být 12b hodnota zarovnána vlevo nebo vpravo podle nastavení v registru AD0CN.

1.2.3 D/A převodník

Používaný procesor má k dispozici dva D/A moduly, které budou oba využity pro generování řídicích signálů. Oba moduly jsou identické, co se týče vlastností i ovládání. Rozsah výstupního napětí modulů je od 0 do $(V_{REF}-\Delta U)$, kde ΔU je hodnota kvantizačního kroku vypočítaná podle (1).

$$\Delta U = U_i - U_{(i-1)} = \frac{U_{ref}}{2n-1} = \frac{U_{ref}}{2^{12}-1} = \frac{2,4}{4095} = \underline{\underline{59mV}} \quad (1)$$

Tento rozsah odpovídá rozsahu hodnot v registrech DACxH a DACxL. Pro oba 12b moduly mohou registry nabývat hodnot od 0 do 4095. Každý z modulů má k ovládání určený SFR registr DACxCON, ve kterém je D/A převod povolen nebo zakázán, dále zde lze nastavit zdroj spouštění D/A převodu od časovačů nebo zápisem do DACxH. Protože z datových registrů, které mají 16 b, je pro D/A převod určeno pouze 12 b, tak je nutné nastavit zarovnání číselné hodnoty v registrech. Takto lze snadno bez předchozích rotací čísel, které zpomalují procesor, určit správnou interpretaci numerické hodnoty uložené v registrech.

1.2.4 Sériové rozhraní UART

Procesor disponuje dvěma nezávislými sériovými linkami typu UART (Universal asynchronous receiver transmitter). V programu bude využívána pouze jedna pro komunikaci mezi přístrojem a ovládací aplikací na PC. Každý UART modul může být nastaven v různých režimech používání. Programem používaný modul UART0 je schopný operovat v plně duplexním asynchronním režimu nebo v poloduplexním

synchronním režimu. Podporovaná je i komunikace v multiprocesorových systémech. V přístroji bude po sériové lince probíhat pouze přenos malého objemu dat. Přístroj bude navíc komunikovat pouze s ovládací aplikací na PC, a proto může pracovat v duplexním asynchronním režimu s pevně nastavenou přenosovou rychlostí. Primárně využíván bude pouze směr dat z PC do přístroje a druhý směr z přístroje do PC bude připraven na případné aplikování softwarového řízení toku nebo na vysílání různých dat směrem k PC v případných pokročilejších verzích přístroje.

UART0, který bude využíván, bude nastaven v režimu „#1“ podle [1]. Tento režim poskytuje plně duplexní asynchronní přenos dat. Využívá 10 bitů na jeden bajt přenášených dat. Osm bitů datových, jeden „startbit“ a jeden „stopbit“. Při přijmutí datového slova se osm datových bitů uloží v SFR registru SBUF0 a „stopbit“ je uložen v registru SCON0.2 v bitu RB80. Přijmutí datového slova může začít kdykoliv po nastavení bitu REN0 (Receive enable bit). Pokud je navíc povoleno přerušení od UART0 a jsou nastaveny bity TI0 a RI0 v registru SCON0, tak při dokončení příjmu datového slova je vygenerováno přerušení, které umožňuje okamžitě odebrat přijaté slovo z registru. Při vysílání dat přenos začíná zapsáním do registru SBUF0. Na konci vysílání je nastaven příznak v registru SCON0.

Přenosová rychlost je v tomto režimu modulu řízena přetečením jednoho z časovačů a pro oba směry ji lze nastavit odděleně, takže přenosová rychlost v jednom směru může být jiná než ve směru opačném. Zdroj generování přenosové rychlosti se volí nastavením příslušných bitů v registru SSTA0. Při zvolení časovače T2 jako zdroje pro generování přenosové rychlosti pro vysílání i příjem je přenosová rychlost dána vztahem (2).

$$UART_Baud_Rate = \frac{T2CLK}{16 \times (65535 - RCAP2)} \quad (2)$$

Kde $T2CLK$ je frekvence hodinového signálu, který je zdrojem pro časovač T2 a $RCAP2$ je reload hodnota pro časovač (viz 2.2.5).

1.2.5 Časovače

V procesoru je k dispozici pět 16b časovačů k všeobecnému používání a jeden Watchdog časovač (časovač hlídající správný chod programu), který nebude programem využíván. Všech pět časovačů je rozděleno do dvou skupin, které mají odlišné vlastnosti a možnosti použití.

První skupina, časovače T0 a T1, jsou pozůstatkem z jádra procesoru 8051. Jsou implementovány jako dvojice 16b registrů, ke kterým se přistupuje pomocí dvou 8b SFR registrů TLx a THx. Oba registry mohou pracovat v režimu časovače, kdy je jejich registr inkrementován na základě hodinového signálu ze systémového hodinového signálu upraveného děličkou kmitočtu nebo v režimu čítače, kdy je jejich registr inkrementován signálem z vnějšího zdroje hodinového signálu. Kromě těchto dvou základních režimů lze zvolit pro tuto skupinu jeden z režimů, které jsou shrnuty v tabulce na obr. 6. Nastavení obou časovačů se provádí pomocí SFR registrů CKCON, TMOD a TCON, ve kterých jsou také aktuální informace o stavu jednotlivých časovačů.

| | 1. skupina: Časovače T0 a T1 | 2. skupina: Časovače T2, T3 a T4 |
|----|---|--|
| 1. | 13b čítač/časovač | 16b čítač/časovač s funkcí „auto-reload“ |
| 2. | 16b čítač/časovač | 16b čítač/časovač s funkcí „Capture“ |
| 3. | 8b čítač/časovač s funkcí „auto-reload“ | Nastavování výstupního pinu |
| 4. | Dva 8b čítače/časovače (Pouze T0) | |

Obr. 6: Tabulka režimů jednotlivých skupin časovačů

Druhá skupina, časovače T2, T3 a T4, je od první skupiny odlišná v několika bodech. Nejdůležitějšími odlišnostmi jsou, že tato skupina dokáže čítat v obou směrech a také umožňuje přímo nastavovat přídružený výstupní pin, čehož se dá využít pro bezobslužné generování obdélníkového signálu na pinech procesoru. Pracovní režimy této skupiny jsou shrnuty v tabulce na obr. 6. Jejich nastavení se provádí přes SFR registry TMRxCF, TMRxCN. V těchto registrech jsou zároveň informace o aktuálních stavech jednotlivých časovačů.

Pro automatické generování přerušení s jistou periodou opakování je využíván režim „auto-reload“, který je implementován odlišně u obou skupin časovačů. Tento režim umožňuje při přetečení časovače vygenerovat přerušení a následně automaticky načíst inicializační hodnotu do registru časovače. Tato „reload“ hodnota musí být uchovávána v registrech k tomu určených, a právě v tomto bodě se liší obě skupiny časovačů. První skupina využívá k čítání v tomto režimu pouze jeden ze dvou registrů časovače (TLx) a druhý registr (THx) využívá pro uchovávání „reload“ hodnoty. Proto první skupina časovačů využívá v režimu „auto-reload“ pouze 8 b. Druhá skupina má pro účely

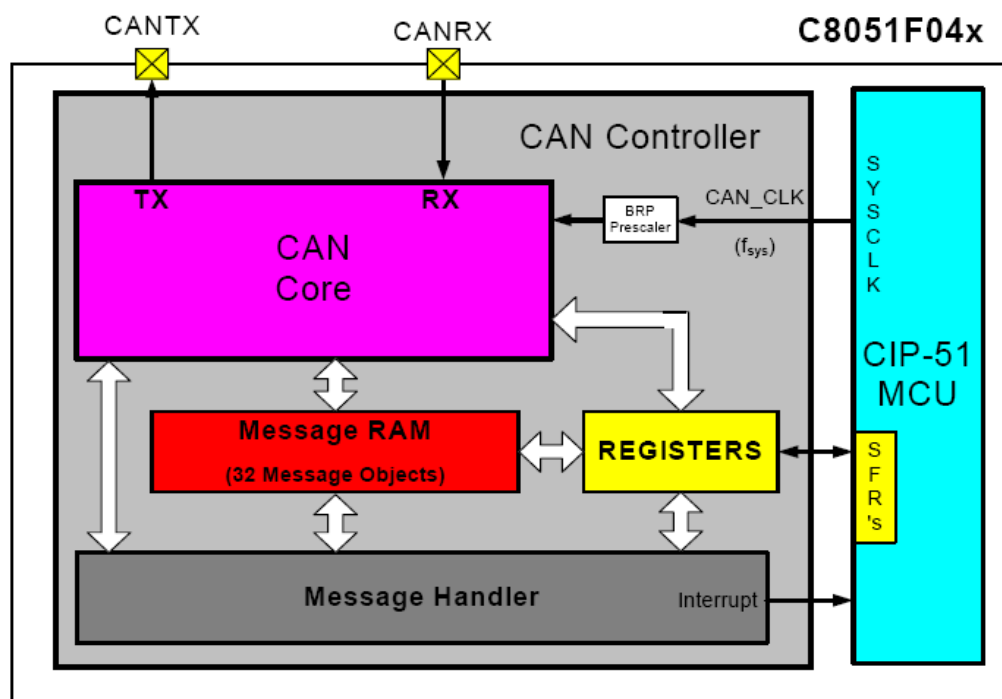
uchování „reload“ hodnoty zvláštní registry RCAPxL a RCAPxH, a proto mohou i v tomto režimu využívat plných 16 b ze svého rozsahu.

Druhá skupina časovačů disponuje také výše zmíněnou vlastností přímo nastavovat přidružený výstupní pin. Každý z druhé skupiny časovačů má určený pin, který lze omezeně nastavit na jiný díky funkci „Crossbar“ (viz [1]). Na tomto pinu je časovač schopný absolutně transparentně generovat obdélníkový průběh nastavené frekvence. Frekvence generovaného signálu se vypočítá podle (3), kde $f_{Txclock}$ je frekvence zdroje hodinového signálu pro daný časovač a $RCAPx$ je hodnota v „reload“ registru daného časovače (viz výše). Této vlastnosti bude přístroj využívat ke generování obdélníkových průběhů na základě aktuální rychlosti a otáček automobilu. Tato funkce nebyla přímo popsána v zadání, ale pro měření testů obsluha tyto informace potřebuje. Protože přístrojem později zamezíme přístup k diagnostické sběrnici, ze které tyto informace obsluha získávala, je potřeba tyto informace poskytnout i nadále.

$$f_{OBD} = \frac{f_{Txclock}}{2 \times (65535 - RCAPx)} \quad (3)$$

1.2.6 Řadič sběrnice CAN

Jednou ze speciálních periférií řady, ze které je používán procesor je zabudovaný modul pro komunikaci na sběrnici CAN 2.0B (Controller area network) viz [2]. Tento modul umožňuje nastavení až 32 objektů zpráv, každý s příslušnou maskou identifikátoru pro filtrování. Modul CAN v procesoru se skládá ze standardního Bosch CAN řadiče (CAN Core), paměti RAM na objekty zpráv, která není namapována v paměti jádra 8051, „Message handler“, který obsluhuje zacházení se zprávami, a z příslušných registrů, které umožňují ovládání celého modulu. Blokové schéma celého modulu a jeho schematické propojení s jádrem procesoru je znázorněno na obr. 7. Celý modul značně ulehčí čtení dat ze sběrnice. Přímé čtení dat ze sběrnice vyloučené není, ale implementovat do kódu protokol sběrnice CAN podle [2] by byl úkol na samostatnou práci. V dalším textu budou součásti modulu CAN pro přehlednost označovány pouze přívlaskem CAN (CAN RAM, CAN řadič, apod.).



Obr. 7: Blokové schéma řadiče sběrnice CAN a jeho připojení k jádru procesoru

Modul CAN je schopný pracovat do přenosové rychlosti až 1Mb/s, což je podle [2] limitní přenosová rychlost pro sběrnici CAN. Teoreticky je na kratší vzdálenosti možné použít i rychlost větší, protože hlavní omezení přenosové rychlosti vychází z času, který trvá signálu zabrat celou délku vedení sběrnice. Pro přenosovou rychlost 500 kb/s, která je používána v automobilu, je tedy maximální délka vedení až desítky metrů.

Každý z 32 objektů použitelných pro komunikaci může být nastaven jako přijímací nebo vysílací. Přijatá data a objekty zpráv, včetně masek jejich identifikátorů, jsou ukládány do paměti zpráv. Všechny funkce protokolu pro odesílání i přijímání a filtrování zpráv vykonávány CAN řadičem a nikoliv procesorem samotným. Tyto procesy tedy probíhají v pozadí programu procesoru a zasahují do něj pouze v situacích, kdy dojde k přijetí hlídané zprávy, což vyvolá přerušení hlavního programu. Procesor přistupuje ke CAN modulu pomocí sady SFR registrů, které má namapované mezi ostatní SFR registry.

CAN řadič v modulu provádí nastavování pinů (CANTX a CANRX), sério-paralelní převod dat o ostatní úkoly spojené s protokolem sběrnice, jako například filtrování příchozích zpráv. CAN RAM slouží k ukládání 32 objektů se zprávami, které jsou přijaty nebo budou odeslány na CAN síť. Registry modulu CAN a „Message Handler“ umožňují spojení mezi modulem CAN a jádrem procesoru 8051.

Rozdělení CAN registrů podle [1]:

- **Registry CAN řadiče** - Registry pro ovládání, nastavení přerušení, kontrolu chybových hlášení, kontrolu stavu na sběrnici a testovacího módu.
- **Sady registrů pro obsluhování objektů zpráv** - Používají se pro konfiguraci 32 objektů zpráv a pro příjem a vysílání dat z nebo do objektů zpráv. Pomocí těchto dvou sad registrů může procesor přistupovat do CAN RAM. Po nastavení čísla objektu zprávy do „COMMAND REQUEST“ registru jedné ze sad (IF1 nebo IF2) se přenese obsah celé sady z nebo do objektu zprávy v CAN RAM.
- **Registry „Message Handleru“** – Tyto registry poskytují procesoru informace o objektech zpráv, jako například příznaky platnosti zprávy (MSGVLD), příznaky nové zprávy (NEW DATA) nebo informace o zdroji přerušení (INTERRUPT PENDING). Všechny tyto registry jsou pouze ke čtení.
- **Registry namapované mezi SFR jádra procesoru** - Tyto registry umožňují přímý zápis do registrů v modulu CAN nebo slouží jako registry pro nepřímý přístup ke všem registrům CAN modulu.

Při přístupu k registrům, které nejsou namapovány mezi SFR procesoru, musí být využito nepřímého přístupu. Každý registr v CAN RAM má svůj index (viz [1], str. 235-236). V CAN RAM je k dispozici 256 B adres, což odpovídá 128 registrům, které lze adresovat pomocí 8b SFR registru CAN0ADR. K datům příslušného adresovaného registru lze následně přistupovat pomocí dvou 8b SFR registrů (CAN0DATH a CAN0DATL). Pokud jde o zápis do registru, tak po zapsání hodnoty do registrů CAN0DATH a CAN0DATL se provede zápis na adresované místo v CAN RAM. Pokud jde o čtení, tak po adresování registru pomocí CAN0ADR může procesor hodnotu tohoto registru přečíst v registrech CAN0DATH a CAN0DATL. Při konfiguraci jednotlivých objektů zpráv je často zapotřebí přistupovat postupně ke všem registrům a nastavovat jejich hodnoty. Pro tyto potřeby zařizuje procesor autoinkrementaci ve dvou oblastech CAN RAM. Při nastavování těchto dvou oblastí pak stačí pouze postupně zapisovat data do datových registrů CAN0DATH a CAN0DATL a po zapsání nižšího bajtu se provede automatická inkrementace adresovacího registru CAN0ADR. Autoinkrementace nastává i v případech, kdy

procesor z oblasti těchto registrů čte. Tyto dvě oblasti pokrývají dvě sady registrů pro obsluhování objektů zpráv, které slouží k přístupu k objektům zpráv (viz výše). První sada IF1 je v CAN RAM umístěna na adresách 0x08-0x12 a druhá sada IF2 na adresách 0x20-0x2A.

Při inicializaci modulu a následné konfiguraci objektů zpráv je nutné dodržet předepsaný postup. Po resetování obvodu jsou všechny registry uvedeny do výchozího stavu, který je definovaný v [2]. Při tomto výchozím stavu je modul nastaven do inicializačního režimu. Tento režim umožňuje nakonfigurovat časování bitu a všech ostatních parametrů potřebných pro správný chod modulu na sběrnici. Pokud je zapotřebí znovu inicializovat nebo měnit základní nastavení CAN modulu, tak je možné inicializační režim vyvolat i softwarově, nastavením bitů CCE a INIT v SFR registru CAN0CN. Tyto bity povolí změnu konfiguračních registrů (bit Change Enable) a uvedou modul do inicializačního režimu (bit INIT). Po těchto nutných podmínkách může následovat sled inicializačních dat směrem do modulu. Obvyklý způsob je začít konfigurací „BIT TIMING“ registru, který je klíčový pro časování akcí na sběrnici. Hodnota tohoto registru určuje přenosovou rychlost na sběrnici, která je odvozena od frekvence systémových hodin procesoru, a zároveň definuje čas odebrání vzorku při čtení bitu. Tento čas je daný několika číselnými hodnotami, které v praxi reprezentují čas před a po odebrání vzorku ze sběrnice a také čas, po který může jádro CAN tento bod posouvat v rámci resynchronizace. Podle [2] není čas odebrání vzorku kritická hodnota a většinou se špatné nastavení v běžné aplikaci neprojeví, přesto je dobré tyto hodnoty nastavit, tak aby byla tolerance co největší a zařízení tak více odolné proti selhání způsobenému nepřesností oscilátoru.

Dále je zapotřebí nakonfigurovat všechny objekty zpráv, které budou v aplikaci využívány. Standardně, když je modul nastaven v režimu „Basic“ podle [2], se k odesílání do CAN RAM využívá sada registrů IF1 a ke čtení z CAN RAM IF2, přičemž operace čtení, vyvolané IF2, mají přednost, takže mohou přerušit IF1. Všechny objekty se zpravidla před konfigurací uvedou do stavu, ve kterém je CAN řadič bude přehlížet. Tato vlastnost objektu se nastaví pomocí vynulování bitu MSGVAL (platnost zprávy) v CAN registru Arb2 pro všechny objekty. Potřebným objektům je pak při další konfiguraci třeba nastavit hodnotu MSGVAL na „1“ a tím dát najevo, že tento objekt bude hlídán. K nastavení objektu zprávy je zapotřebí definovat identifikátor, filtr identifikátoru, směr zprávy, povolení resp. zakázání přerušení od příslušné zprávy, formát rámce zprávy a počet datových bajtů ve zprávě. Všechny tyto parametry se pro

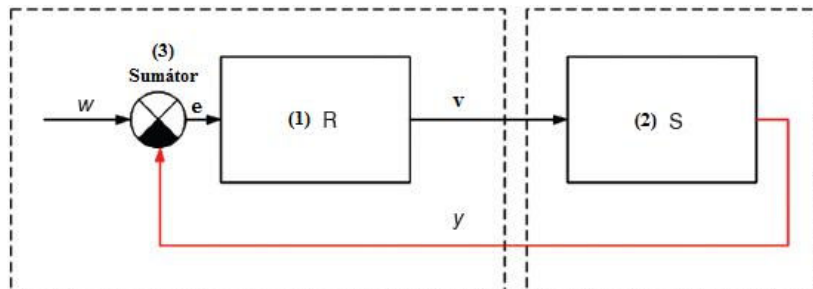
každou zprávu zvlášť nastavují ve dvou arbitrážních registrech. Po provedení konfigurace objektů lze modul CAN přepnout z inicializačního režimu vynulováním bitů CCE a INIT v CAN0CN registru v procesoru. Okamžikem vynulování těchto bitů začne modul pracovat a monitorovat zprávy na sběrnici podle nastavení objektů. Proces konfigurace je možné provést i v jiném pořadí, protože při nastavení „BIT TIMING“ registru nedochází ke změnám v objektech zpráv, takže konfiguraci objektů lze vykonat už v prvotní inicializaci procesoru.

Když CAN modul objeví na sběrnici zprávu, kterou má v objektech nastavenou jako přijímací a která má generovat přerušení, tak ji přijme, uloží do CAN RAM a po nastavení příznaků vygeneruje přerušení. Tímto způsobem je procesor schopný registrovat všechny kontrolované zprávy na sběrnici a v okamžiku jejich přijetí jejich data okamžitě načíst do své paměti RAM. Proces načtení přijaté zprávy z CAN RAM je stejný jako při zapisování datových bajtů při konfiguraci CAN modulu. Jediný rozdíl je v tom, že je ke čtení použita druhá sada registrů pro obsluhování objektů zpráv (IF2). Základní jednoduchý postup je tedy při přerušení načíst registr CAN0STA, ve kterém je uložen status modulu. Z tohoto registru lze zjistit, zda bylo přerušení vyvoláno přijatou nebo odeslanou zprávou, nebo chybou na sběrnici. Pokud však v programu není povoleno jiné přerušení než od přijímané zprávy, tak tento krok může být vynechán. Dalším krokem je načtení CAN registru INTREG, ze kterého v této situaci lze vyčíst číslo zprávy, která přerušení vyvolala. Vyčtení samotných dat zprávy, jejíž číslo bylo zjištěno, proběhne zapsáním čísla zprávy do „COMMAND REQUEST“ registru. Následným adresováním datových registrů přijímací sady IF2 (IF2DATA1, IF2DATA2, atd.) procesor postupně vyčte až všech osm datových bajtů, které je možné následně upravovat a ukládat v paměti RAM.

1.2.7 Implementace zpětnovazební řídicí smyčky

Aby bylo možné přístroj v automatickém režimu používat na různých typech automobilů je potřeba ošetřit odlišnosti agregátů jednotlivých typů automobilů a výkonů. Na vyřešení se nabízí několik variant. Nejjednodušší varianta pro programování by byla umístit postupně všechny automobily, pro které bude zařízení určeno, na válce, na kterých bude později přístroj sloužit, a naměřit u všech automobilů závislost rychlosti na napětí plynového pedálu. Vzniklá tabulka hodnot by se dala zakomponovat do PC aplikace, která by následně vyřešila veškeré početní úlohy a do

procesoru by se přenášela pouze datová řada, která by byla úměrná průběhu výstupního napětí. Toto řešení bylo původním záměrem, ovšem po důkladnějším prozkoumání možností a vlastností automobilů se ukázalo jako nevhodné. Nejen že by práce vynaložená naměřením všech tabulek pro všechny typy automobilů zabrala dlouhé hodiny, ale v praxi to není ani možné, protože všechny vozy pohromadě k dispozici nejsou. Toto řešení by bylo navíc závislé na použití stejných válců, respektive na stejné síle, kterou jsou kola na válcích brzděna. Při novém typu automobilu by se navíc tabulka v aplikaci musela doplňovat, a to dělá toto řešení značně neflexibilní. Druhé řešení, které se zdá jako optimální vzhledem k možnostem, které automobil poskytuje, je řídit rychlost automobilu, která je hlavní sledovanou veličinou, ve zpětnovazební smyčce s použitím informací o rychlosti a otáčkách. V procesoru by tedy vystupovala jako hlavní žádaná hodnota rychlost automobilu a v některých situacích otáčky motoru. Automobil, respektive motor automobilu s jeho řídicí jednotkou, by pak vystupoval jako regulovaná soustava a plynový pedál, respektive zapojený přístroj, jako regulátor a částečně jako akční člen. Na obr. 8 je zobrazeno základní schéma regulačního obvodu a dále je popsána jeho aplikace na řešený případ.



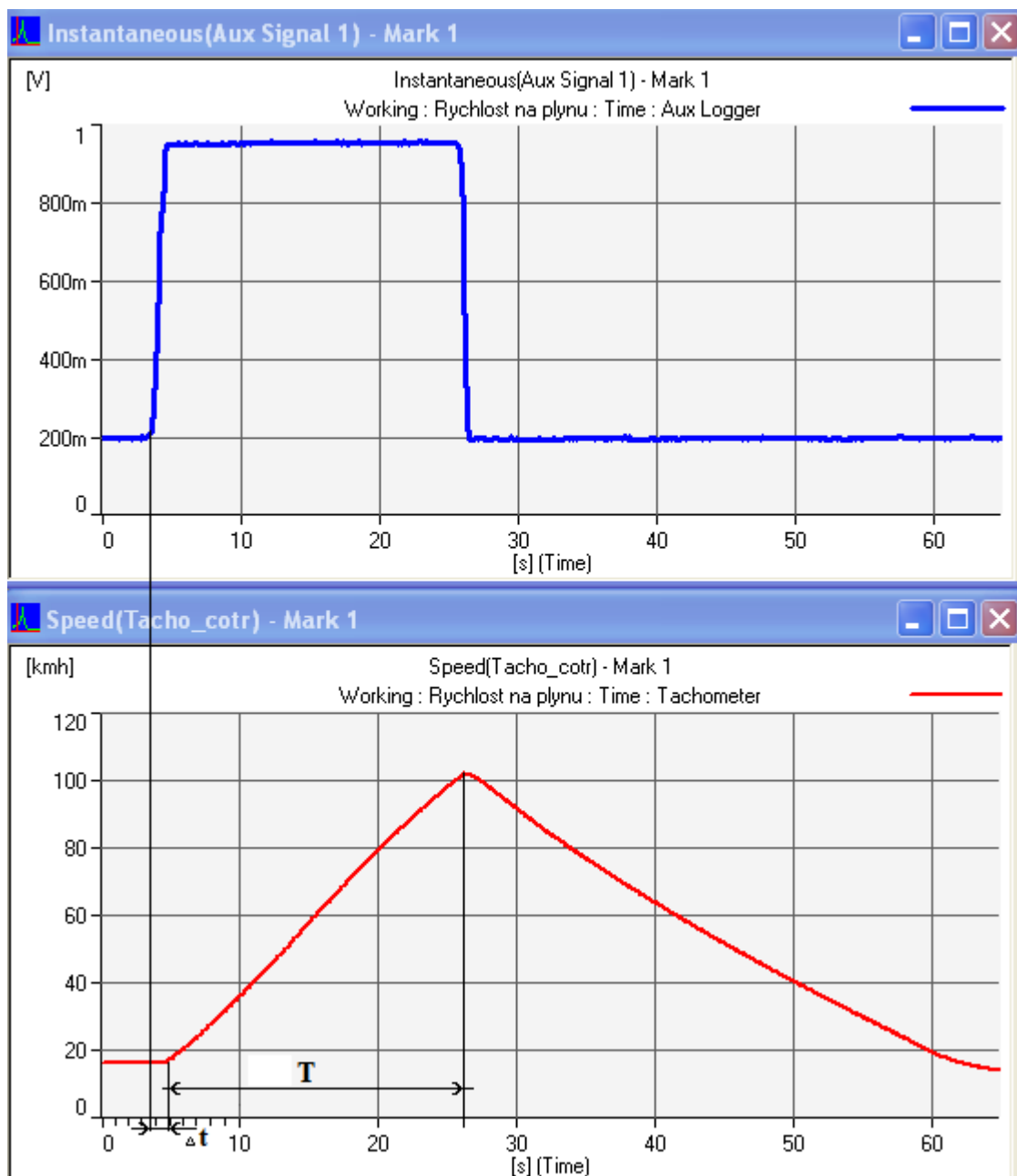
Obr. 8: Základní regulační obvod

V základním regulačním obvodu vystupující veličina w je žádaná hodnota regulované veličiny. V konkrétním řešeném případě jde o aktuální žádanou hodnotu rychlosti, která vychází z požadovaného časového průběhu rychlosti. Ten si uživatel zvolí pomocí parametrů testu v ovládací aplikaci. Žádaná hodnota vstupuje do sumátoru (3), který je již v konkrétním řešeném případě realizován softwarově v kódu procesoru. Sumátor má za úkol přijímat signál žádané hodnoty w a porovnávat ho, respektive odčítat od něj aktuální hodnotu regulované veličiny y . Regulovaná veličina je v řešeném případě získávána z komunikační sběrnice v automobilu. Odečtením veličin w a y získáme veličinu e ($e=w-y$). Tato veličina, nazývaná regulační odchylka, se reálně

v systému nevyskytuje. Vystupuje pouze jako proměnná v programu procesoru. Tuto odchylku zpracovává regulátor (1), který z regulační odchylky e vypočítá „optimální“ regulační zásah. Regulační zásah je v regulované soustavě označen jako veličina v . Někdy se tato veličina nazývá také akční veličina a v systému je realizována pomocí napěťových úrovní, které vystupují z přístroje. Tento signál vstupuje do regulované soustavy, kde působí tak, aby ovlivnil regulovanou veličinu. Regulovanou soustavu si lze představit jako kombinaci motoru a jeho řídicí jednotky, k níž je potřeba přistupovat jako k celku, jehož vnitřní strukturu nelze měnit. Zbývá ještě popsat vznik regulované veličiny. Ta je v automobilu měřena z několika zdrojů, což ale pro účely programu není podstatné. Podstatné je, že hodnota regulované veličiny je k dispozici a periodicky aktualizována na sběrnici CAN.

V regulačním obvodu na obr. 8 má regulovaná soustava (2) i regulátor (1) konkrétní přenos. Stejně tak je potřeba při řízení znát alespoň přibližně přenos regulované soustavy jako kombinace motoru a řídicí jednotky a vědět, jakým způsobem lze ovlivňovat přenos regulátoru a tím ovlivňovat kvalitu regulačního pochodu.

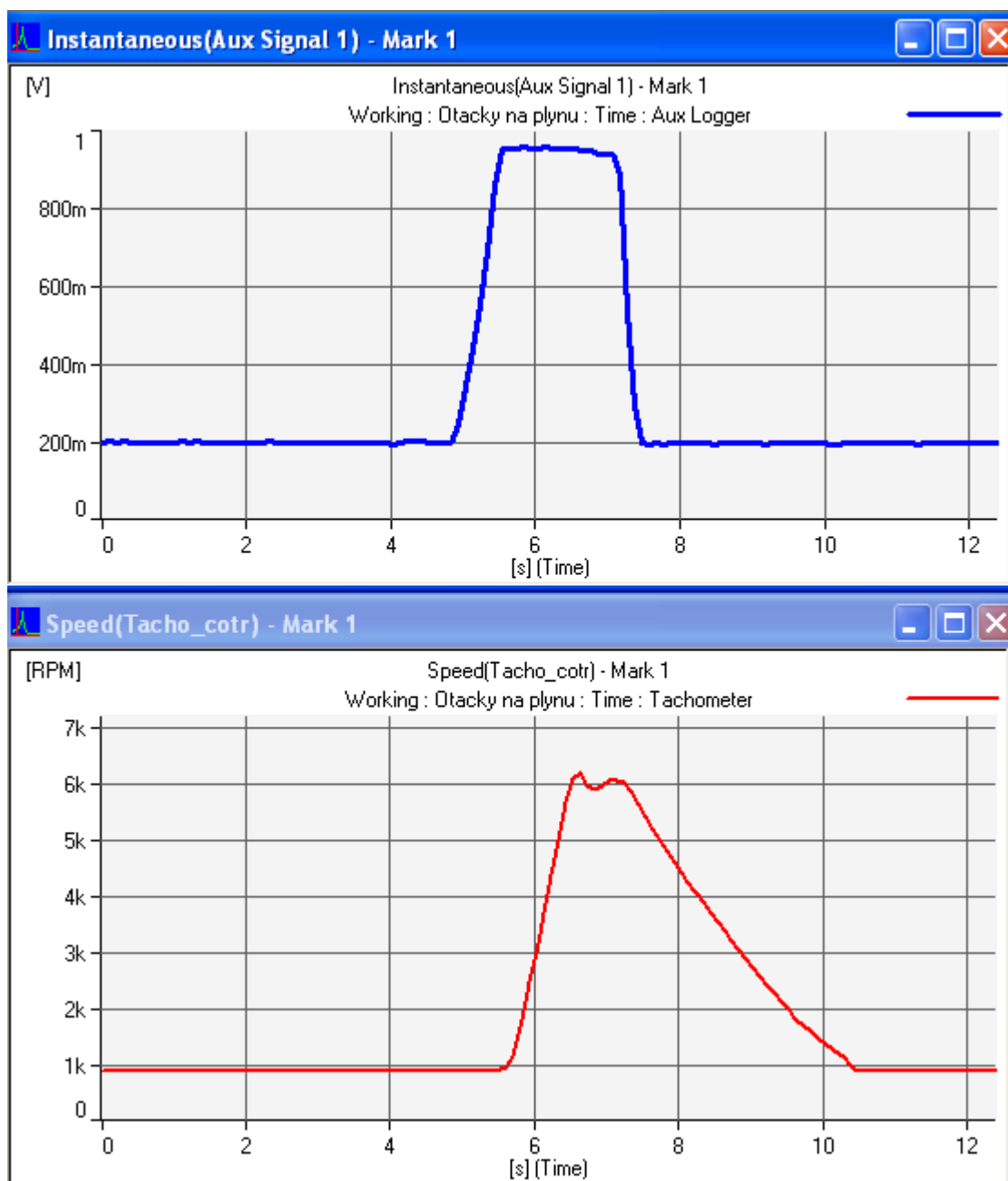
Pro účely zjištění přenosu regulované soustavy byl vytvořen program (viz 2.2.8), který regulovanou soustavu budil jednotkovým skokem na akční veličině. Hodnota tohoto skoku byla zvolena tak, aby motor vybudila z volnoběžných otáček po otáčky maximální. Zjišťovaný přenos lze přibližně určit ze závislosti rychlosti automobilu na jednotkovém skoku řídicí veličiny což je obdoba přechodové charakteristiky soustavy. Tato závislost je zobrazena na obr. 9. Místo přesného určování přenosu soustavy numerickými metodami se lze pro účely této nenáročné regulace spokojit s určením základních parametrů soustavy. Soustavu lze podle přechodové charakteristiky aproximovat soustavou prvního řádu, tedy s jednou časovou konstantou. Časová konstanta soustavy je na obr. 9 naznačená jako hodnota T . Z obrázku je patrné, že časová konstanta je poměrně velká, $T = 22$ s. Řídicí jednotka, jak je z obrázku také vidět, velmi výrazně ovlivňuje celou odezvu motoru a přechodovou charakteristiku velmi výrazně linearizuje. Tato skutečnost je logická s ohledem na reálné zapojení s pedálem, který řidič ovládá nohou. To vysvětluje i výrazný časový posun signálů nazývaný též jako dopravní zpoždění soustavy, který je v obr. 9 označeno jako hodnota Δt .



Obr. 9: Přechodová charakteristika motoru s řídicí jednotkou pro rychlost

Při automatickém režimu přístroje je zapotřebí kromě rychlosti v některých okamžicích regulovat i otáčky motoru. Pro otáčky byla naměřena stejná přechodová charakteristika jako pro rychlost. Tato charakteristika je zobrazena na obr. 10. Hlavním poznatkem z porovnání těchto dvou měření je poměrně rychlá odezva otáček motoru na skok v akční veličině. Tato naměřená závislost, zobrazená na obr. 10 byla ovšem naměřena na automobilu bez zařazeného rychlostního stupně, tudíž motor naprázdno. Charakteristika se zařazeným rychlostním stupněm by byla obdobná jako závislost

rychlosti na obr. 9, protože mezi rychlostí a otáčkami je při jednom zařazeném rychlostním stupni konstantní poměr.



Obr. 10: Přechodová charakteristika motoru s řídicí jednotkou pro otáčky

Všechny výše uvedené výsledky dostatečně popisují soustavu, která bude řízena a lze z nich vyvodit několik základních poznatků, které budou aplikovány při implementaci numerické řídicí smyčky. Jelikož je soustava pro regulaci rychlosti prvního řádu s relativně velkou časovou konstantou, tak pro účely regulace lze implementovat pouze PI regulátor (proporcionálně-integrační) respektive jeho numerickou obdobu, což je regulátor PS (proporcionálně-sumační). Druhým poznatkem je nutná opakovací

frekvence řídicí smyčky. Protože procesor není schopný pracovat v několika úlohách zároveň, je nutné program napsat tak, aby se regulační smyčka opakovala s určitou periodou. Tato perioda je převrácená hodnota výše zmíněné opakovací frekvence řídicí smyčky a je obdobou vzorkovací frekvence v běžných regulátorech. Vzhledem k časové konstantě a dopravnímu zpoždění regulované soustavy není nutné a dokonce ani možné, aby tato frekvence, respektive perioda, byla větší než polovina času dopravního zpoždění soustavy. I když by procesor byl schopný se svými parametry tuto frekvenci, respektive periodu, realizovat řádově v desítkách mikrosekund, tak s regulovanou soustavou těchto parametrů je zapotřebí tuto periodu snížit až na stovky milisekund aby bylo dosaženo optimálních výsledků při regulaci. Tento předpoklad byl ověřen pokusem. Výsledný přenos regulátoru, který je realizovaný numericky uvnitř procesoru, je tedy ovlivnitelný třemi konstantami. Jedná se o implementaci PS regulátoru, kde koeficient I ve vztahu (4) reprezentuje koeficient integrační složky. Druhý koeficient, vystupující ve vztahu (4), je konstanta proporcionální složky P . Třetí konstantou, kterou lze ovlivnit chování regulátoru je výše zmíněná opakovací frekvence regulační smyčky. Tyto hodnoty byly pro testování přístroje určeny experimentálně. Vztah, který je implementovaný v kódu, pro výpočet regulačního zásahu je vidět ve vztahu (4).

$$v_t = I.v_{(t-1)} + K.e = I.v_{(t-1)} + K.(w - y) \quad (4)$$

Při aplikaci řídicí smyčky v procesoru je zapotřebí brát v úvahu reálné parametry soustavy a regulačních veličin. Při určitých hodnotách žádané a regulované veličiny může snadno dojít ke stavu, že regulační odchylka bude záporná a za jistých podmínek se může stát, že i výsledná akční veličina v_t vyjde záporná. Takovouto situaci by přístroj nebyl schopný řešit. Nejen že D/A převodník není schopný generovat záporná napětí, ale takovéto napětí by nebyla schopná zpracovat ani řídicí jednotka na vstupu regulované soustavy, protože je přizpůsobená na napěťové úrovně popsané v 2.1. Se znalostí vlastností regulované soustavy lze však snadno omezit pracovní rozsah pro přístroj, respektive pro D/A převodník na jeho výstupu. Pokud tedy akční veličina vychází pod tímto povoleným pásmem, tak je nutné její hodnotu nahradit hodnotou nízkou, ovšem v rozmezí povolených napětí. Při akční veličině nad povolené pásmo lze akční veličinu obdobně nahradit maximální povolenou hodnotou.

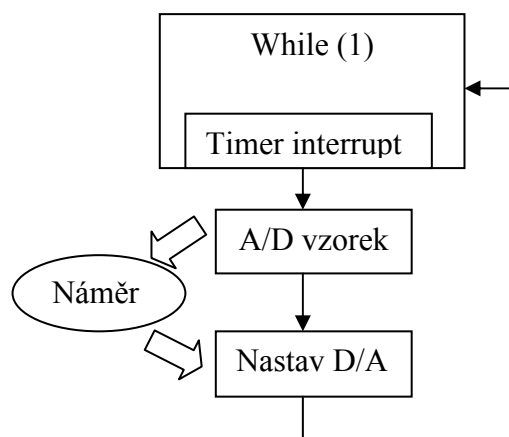
Pokud motor dosáhne horní hranice otáček, které jsou také limitujícím parametrem, nelze dále zvyšovat napětí na vstupu regulované soustavy. V takové situaci se nabízejí dvě možnosti. Nastavit volnoběžné otáčky a celý test ukončit nebo držet vozidlo

v maximálních otáčkách a čekat s ukončením na ovládací tlačítko. V současné verzi programu přístroje je implementována verze s automatickým ukončením testu.

1.2.8 Testovací programy

Pro účely seznámení s automobilem je důležité zjistit, jak bude automobil fungovat jako regulovaná soustava. Pro tyto účely vznikla na začátku vývoje sada testovacích programů, na kterých bylo ověřeno fungování základního návrhu obvodu a odezvy automobilu na elementární řídicí signály. Z těchto vyzkoušených fragmentů kódu později vychází konečný program pro procesor. Všechny zde popsané kódy testovacích programů, jsou přiloženy na CD ve složce „Testovací programy“.

Pro ověření fungování klidového režimu, kdy se bude přístroj chovat transparentně a pouze kopírovat stav ze vstupu na výstup, byl vytvořen program „GAS_ONLY“. Základní funkce programu byla na vstupu číst data na A/D převodníku a nastavovat je na výstup pomocí D/A převodníků. Na realizaci se nabízejí dvě možnosti. Čtení a zápis dělat v pevně určených časových krocích, které budou generovány pomocí přerušení od časovače nebo konverzi signálu dělat v nekonečné smyčce bez ohledu na čas. Program nakonec využíval první variantu z několika důvodů. Řešení v nekonečné smyčce bez generování přerušení by mělo za následek neustálé vytížení procesoru. V tomto jednoduchém případě by to nevadilo, ale později, kdy se procesor bude muset v reálném čase vypořádat se zpětnovazebním řízením regulované soustavy a vyčítáním informací ze sběrnice CAN, se musí časový sled událostí lépe plánovat. Dalším důvodem, kromě plýtvání systémovými prostředky, je potřeba časové informace v systému. Ze zadání je jasné, že hlavním úkolem přístroje je kontrolovat a řídit zrychlení automobilu, což je fyzikální veličina závislá na čase. Bez přítomnosti časové informace v programu by tedy program nebyl schopný tento nejdůležitější parametr kontrolovat. Na obr. 11 je základní vývojový diagram implementovaný v programu „GAS_ONLY“. Při testování tohoto programu na reálném automobilu byl jako zdroj napětového signálu pro A/D převod použit pouze potenciometr, takže otáčky automobilu byly nastavitelné potenciometrem. Při prvním používání tohoto programu vyšel najevo nedostatek, který přístroj má. Přístroj musí být připojen a zapnut dříve než dojde k nastartování vozidla (viz 1.1).

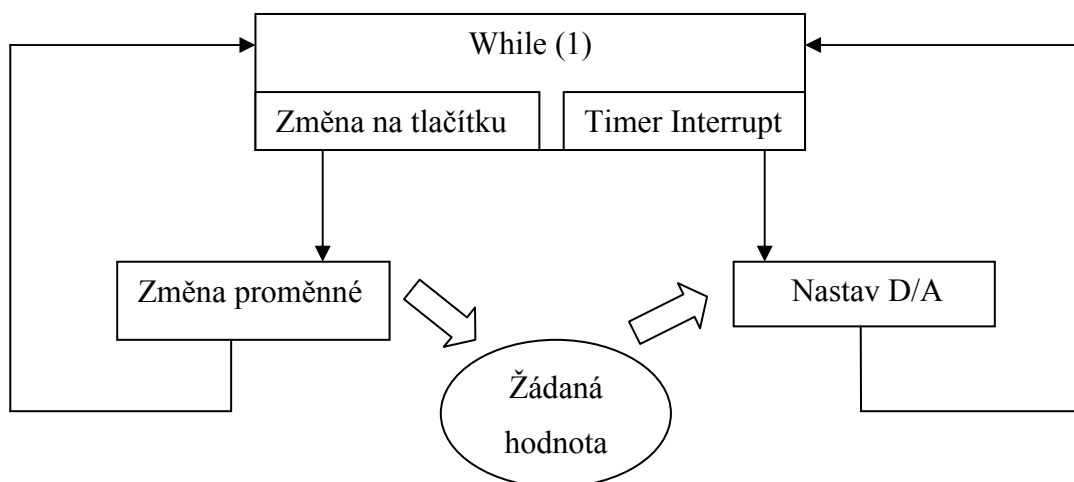


Obr. 11: Vývojový diagram programu „GAS_ONLY“

Pokud bude vozidlo nastartováno a přístroj na vstup řídicí jednotky nebude generovat správná napětí, řídicí jednotka nahlásí chybu a přepne se do režimu zvýšených volnoběžných otáček. V takovém stavu pak otáčky automobilu nelze řídit, i když přístroj začne generovat správný signál. Po dalším spuštění se chyba sama vymaže, takže lze otáčky v pořádku řídit i po předchozí chybě. Při prvních testech je vidět, že vozidlo reaguje na změnu na potenciometru poměrně pomalu. Z toho lze usoudit, že nebude potřeba generovat signál s velkou frekvencí, protože soustava je poměrně pomalá. Aby byla tato úvaha podložena měřením, je potřeba zjistit, jak rychle automobil reaguje na změny řídicích signálů. Pro účely tohoto měření byl vyvinut jednoduchý program, který bude popsán dále.

Druhým programem ze základní testovací sady, která byla vytvořena na začátku, je program „GAS_MANUAL“. Tento program je upravená verze programu „GAS_ONLY“. Smysl tohoto programu je generovat digitální řídicí signál uvnitř procesoru a z něho dopočítávat proměnné pro nastavení na D/A převodnících. Zatímco v prvním testovacím programu byla hodnota z A/D převodníku přímo hodnotou pro D/A převodník, v tomto případě byla ve statické proměnné uložena informace o jakési žádané hodnotě, kterou bylo možné upravovat pomocí tlačítka. Na začátku programování byla pro jednoduchost hodnota do D/A převodníku zapisována pouze do horního bajtu, jak se ale při testu ukázalo, tak při dopočítávání druhé hodnoty napětí, která by měla ideálně být přesným dvojnásobkem hodnoty první, docházelo k výrazné chybě, která v některých okamžicích způsobovala chybové hlášení. Tyto chyby byly vyeliminovány po rozšíření přístroje o koncový stupeň. Program tedy generuje jen jeden signál a ten je při kvantizačním kroku, daném vztahem (1), už dostatečný k tomu, aby

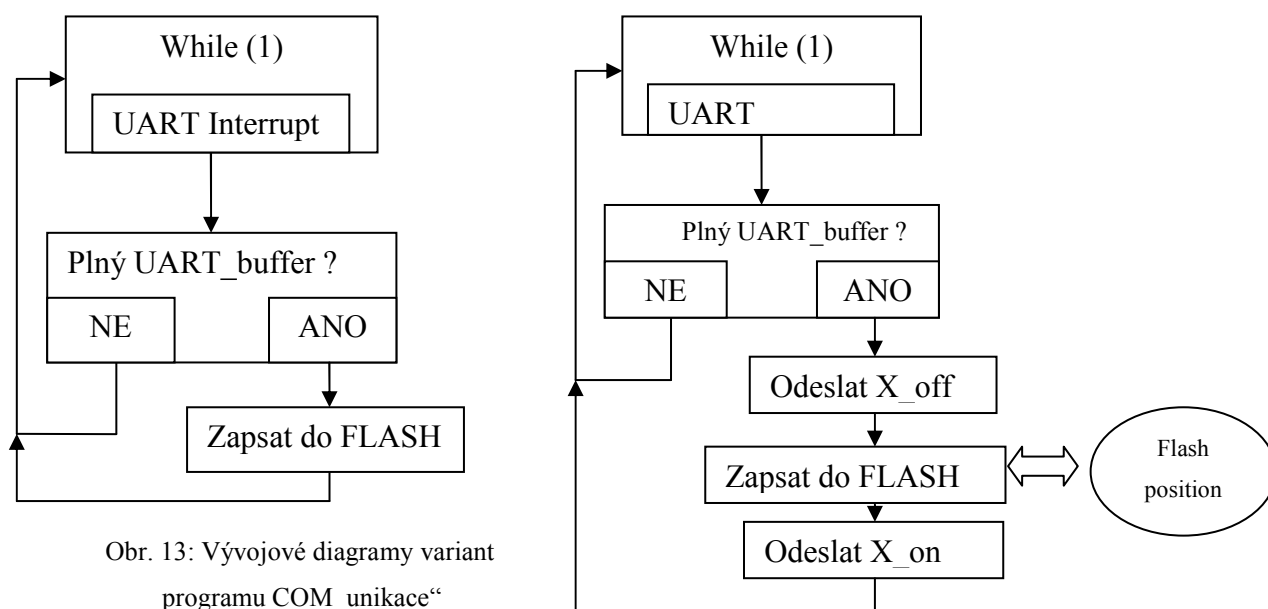
splnil nároky na toleranci napětových signálů. Na obr. 12 je základní vývojový diagram programu.



Obr. 12: Vývojový diagram programu „GAS_MANUAL“

Třetí testovací program byl napsán pro ověření základních funkcí přenosu po sériové lince. Na vývojové desce není zapojený převodník z USB na RS232, který bude v konečné podobě v obvodu přístroje zařizovat konverzi mezi signály, proto byl pro vyzkoušení použit běžně dostupný kabelový převodník USB/RS232. Program měl za úkol nastavit potřebné parametry UARTu v procesoru a vyzkoušet příjem dat ze strany PC. Na PC byl k odesílání dat použit terminál, kterým byla data vysílána na virtuální port, na kterém byl připojen procesor. Cílem tohoto zkoušení bylo zjistit, zda je procesor schopen bezpečně přijmout data ze sériové linky při zvolené přenosové rychlosti a zda je schopný přijmout větší balík dat i bez hardwarového nebo softwarového řízení toku na sběrnici. Integrovaný převodník signálů, který bude v konečné verzi zapojený v obvodu, sice plně podporuje všechny druhy řízení toku, ale pro jednoduchost programu i zapojení je výhodnější pokud se přenos bez řízení toku obejde. Dalším důvodem proč řízení toku neaplikovat je fakt, že objem dat, která budou později touto cestou vysílána z PC do procesoru, bude natolik malý, že ho bude možno přijmout najednou a tudíž bez nutnosti řízení toku. Při testování programu se potvrdily předpoklady, že procesor je natolik výkonný, aby stačil obsloužit přenos dat na sběrnici a pokud data při přenosu není nutné ukládat rovnou do paměti FLASH, tak řízení toku není potřeba aplikovat. Při větších souborech dat, kdy už procesoru nestačí zásobník, který mu byl softwarově přidělen a je nucený data při přenosu ukládat do paměti FLASH, dochází k problémům. Zápis do paměti FLASH v procesoru je na poměry na

sběrnice pomalá procedura, a proto bez řízení toku dochází ke ztrátě dat na sběrnici. Do budoucích verzí programu se tedy nabízí dvě základní varianty jak přenos po sériové lince obsloužit. První varianta je jednodušší, bez nutnosti aplikace řízení toku. Při tomto způsobu, který je prozatím pro vyvíjenou aplikaci preferovaný, nastávají ovšem jistá omezení. Vysílaná data ze strany PC musí mít maximální velikost stejnou, jako má softwarově omezený přijímací zásobník definovaný v procesoru. Větší objem dat nelze se zárukou přijmout. Druhá varianta využívá řízení toku na sběrnici. Nejvhodnějším řešením je softwarové řízení toku, které nevyžaduje odlišné zapojení než první varianta, takže tyto dvě varianty mohou být při vývoji programu pružně zaměňovány podle potřeb aplikace. Na obr. 13 je vývojový diagram obou variant programu „COM_unikace“ pro testování přenosu po sériové lince.



Jelikož je v průběhu práce zapotřebí znát alespoň přibližně parametry regulované soustavy, byl pro tyto účely sestaven program, který tyto parametry zjistí. Program byl pracovně nazván „Simple_Auto“. Jeho úkolem je na vstupu regulované soustavy vygenerovat jednotkový skok akční veličiny a zaznamenat odezvu regulované soustavy. Program funguje na jednoduchém principu, kde pomocí časovače měří čas a po určitých časových krocích, které jsou v této verzi programu ještě pevně dány programem, mění hodnotu akční veličiny na výstupu. Tuto hodnotu program vyčítá z FLASH paměti, do které se nejdříve data pošlou po sériové lince. Hodnoty uložené v paměti reprezentují napětíovou úroveň na pedálu. Ve fázi testování tohoto programu ještě nebyla odladěna komunikace se sběrnici CAN v automobilu, a proto nebylo možné použít zpětnou vazbu

s informacemi o rychlosti, které by bylo možné zaznamenávat. Napětíové úrovně byly proto zvoleny experimentálně tak, aby přibližně odpovídaly skoku akční veličiny z volnoběžných otáček po otáčky maximální pro testovaný typ automobilu. Pro záznam průběhů akční a regulované veličiny byla využita platforma „Pulse“, která je využívána pro multifunkční počítačové měření v laboratoři, pro kterou je přístroj konstruovaný.

1.2.9 Finální verze programu pro procesor

Základní funkce a princip programu pro procesor jsou již popsány v průběhu předchozích kapitol. V této kapitole bude důkladněji rozebrána realizace všech potřebných funkcí přístroje. Text této kapitoly se vztahuje ke zdrojovému kódu a uceluje informace o jeho principu. Celý kód včetně komentářů je na přiloženém CD nazván jako „FinalniVerze.C“.

V základním režimu „Gas only“ je aktivován A/D převodník, který měří signál na reálném pedálu. Tato zjištěná hodnota je poté nastavována na výstup pomocí D/A převodníku. Měření i nastavování je spouštěno cyklicky pomocí přerušení od časovače T0. Ten je nastaven, aby generoval přerušení v intervalu 0,5 ms. Tato hodnota není náhodná. Její důvod bude objasněn později. V nekonečné smyčce programu jsou testována tlačítka, která slouží pro obsluhu přístroje. Při stisknutí tlačítka, které bude trvat déle, než je maximální čas obsluhy všech přerušení používaných v této části programu, bude v hlavním programu toto stisknutí zaručeně zaznamenáno. S ohledem na aktuální režim přístroje bude provedena příslušná změna režimu. Při změně režimu vyvolané tlačítkem dojde však pouze ke změně v proměnné, která reprezentuje režim přístroje. Změna režimu se projeví až při dalším přerušení, které je cyklicky generováno časovačem T0.

Výše zmíněný interval je v automatickém režimu používán jako základní časový údaj, podle kterého jsou vykonávány všechny úkoly tohoto režimu. Těchto úkolů je několik a každý má primárně jinou frekvenci opakování. V automatickém režimu je nutné provádět cyklicky regulační výpočty. Tyto výpočty jsou v programu prováděny vzhledem ke skutečnostem popsaným v 2.2.7 pouze každých 500 ms. Dalším úkolem automatického režimu je cyklicky inkrementovat žádanou hodnotu rychlosti. Hodnota inkrementu i hodnota jeho časování jsou do přístroje předávány z ovládací aplikace a v procesoru jsou uchovávány ve flash paměti, dokud nejsou přepsány novým programem. Posledním úkolem automatického režimu je blikat s diodou, což signalizuje běh programu. Tento proces je vyvolán každých 500 ms, takže dioda bliká s frekvencí

1Hz. Automatický režim je po nastavení uveden do stavu, kdy ještě program není spuštěn. V tomto stavu přístroj reguluje otáčky automobilu na hodnotu volnoběžných otáček. Tato hodnota je přístroji opět předávána z ovládací aplikace, takže ji lze volit jako parametr testu. V tomto režimu kromě regulační smyčky neprobíhají žádné další cyklické úkoly. Výše zmíněné akce jako blikání a inkrementace žádané hodnoty probíhají až při spuštění běhu programu tlačítkem „Start/Stop“.

Všechny akce jsou vykonávány nebo volány z přerušovací rutiny, která je vygenerována každých 0,5 ms. V přerušovací rutině program nejdříve zkontroluje, v jakém je přístroj stavu, a podle toho vykoná potřebné operace. Odlišné frekvence vykonávaných akcí jsou v programu řešeny pomocí proměnných, které se při přerušení dekrementují, a při dosažení nuly v příslušné proměnné se vykoná daná akce. Takto je možné s přerušením od jednoho časovače obsluhovat několik akcí s rozlišeným časováním. Akce jako inkrementování žádané hodnoty nebo blikání diodou jsou prováděny přímo v přerušovací rutině, ale regulační výpočty jsou kvůli svojí složitosti volány jako samostatná procedura.

Regulační procedura je společná pro regulaci otáček i rychlosti. Program na začátku regulační procedury zjistí podle aktuálních žádaných hodnot, zda je potřeba regulovat otáčky nebo rychlost. Následně se v regulační proceduře provedou výpočty. Je vypočtena regulační odchylka a akční zásah podle vztahu (4). Na konci procedury je ověřeno, jestli akční zásah nepřekročil povolené pásmo, které je determinováno hodnotami reálného pedálu. Pokud ne, akční zásah je nastaven na výstup pomocí D/A převodníku. Pokud je toto pásmo překročeno, je regulační zásah upraven podle předpokladů uvedených v 2.2.7. Při regulaci rychlosti je současně kontrolována hodnota otáček a při dosažení maximálních otáček je běh programu ukončen.

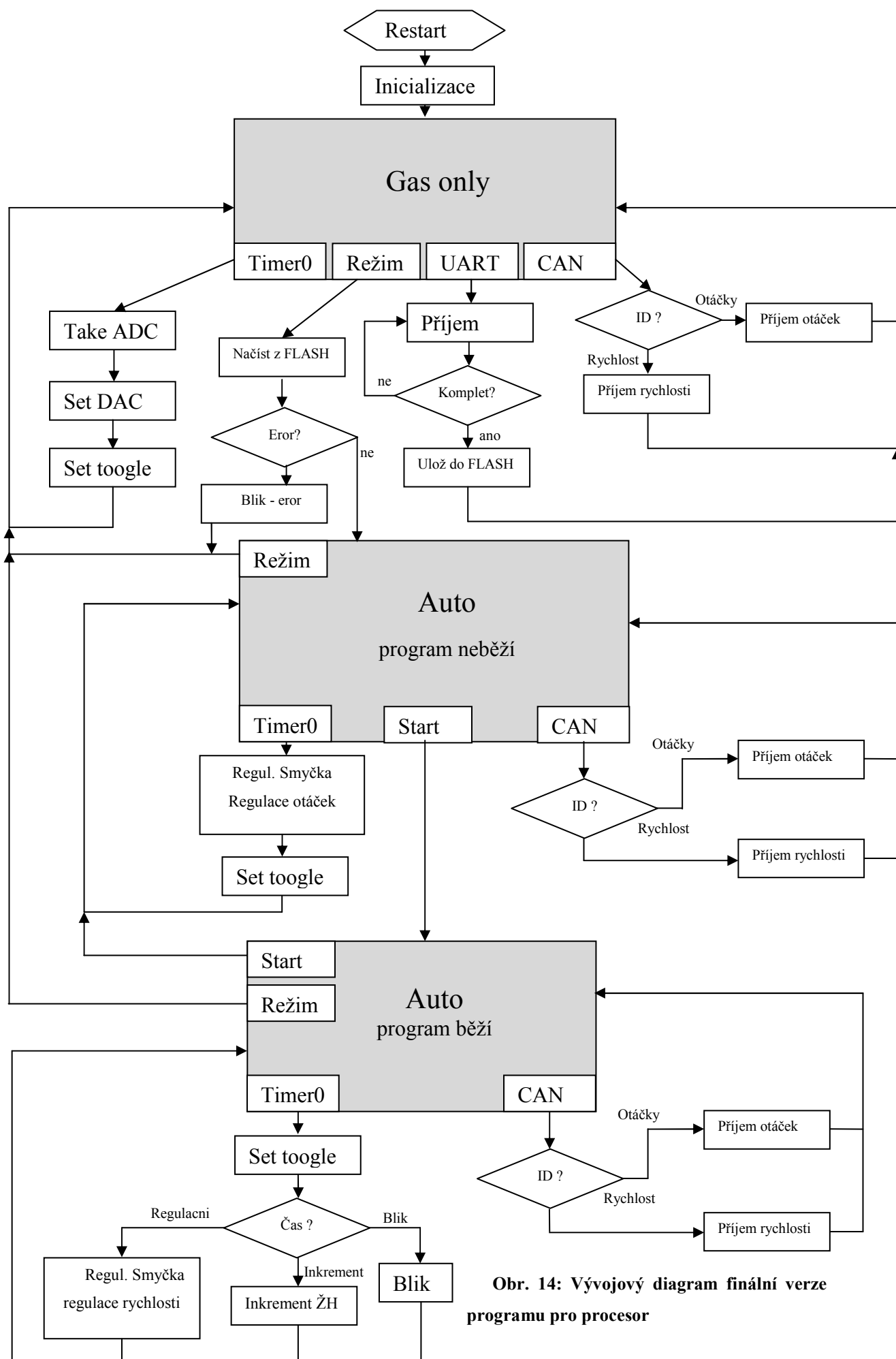
Další funkcí programu je přijetí programovacího balíku dat od ovládací aplikace. Pomocí modulu UART je monitorován stav na sběrnici a při příchozím datovém slovu je vygenerováno přerušení. Toto přerušení je z praktických důvodů zakázáno během automatického režimu. Přímo v přerušovací rutině je vyčteno přijaté datové slovo a uloženo do přijímacího zásobníku. Hodnota ukazatele stavu zásobníku je inkrementována při každém dalším slově, až dosáhne maximální, respektive stanovené, hodnoty. Tato hodnota odpovídá počtu bajtů v datovém rámci (viz 3.3). Pokud je tedy celý rámec přijatý a uložený v zásobníku, je překopírován do flash paměti. Tam je uložen až do příchodu dalšího kompletního balíku programovacích dat. Z paměti flash je balík dat vyčten do jednotlivých konstant potřebných pro regulaci při přepnutí do

automatického režimu. Při načtení dat je také provedena kontrola, zda je ve flash řádně uložený program. Pokud ne, není možné automatický režim spustit. Tímto způsobem je zabráněno řízení připojeného automobilu při nedefinovaných řídicích konstantách.

Speciálním režimem přístroje, který je doplňkový ke dvěma základním režimům, je kalibrační režim. Do tohoto režimu je možné přístroj uvést dlouhým stiskem tlačítka „Režim“. Tato situace je detekována při cyklické kontrole tlačítek. V tomto režimu jsou všechny funkce přístroje omezeny a jediným úkolem tohoto režimu je nastavit na výstup z přístroje pomocí D/A převodníku definovanou napěťovou úroveň a držet ji konstantní. Celý tento úkol je pro svou programovou jednoduchost přímo vykonán při cyklické kontrole tlačítek. Z kalibračního režimu lze přístroj přepnout pouze tlačítkem „Start/Stop“.

Při cyklické kontrole tlačítek jsou softwarově potlačeny zákmity na tlačítku. Protože jsou v přístroji použita běžně dostupná a levná tlačítka, jsou na nich zákmity poměrně značné. Zámit je omezen prodlevou mezi dvěma opakovanými vzorky. Tato prodleva by měla být optimálně nastavena na nejdelší možný čas, po který zákmity na tlačítku probíhají. Experimentálně zvolená hodnota tento čas pravděpodobně několikrát převyšuje, což běh programu nijak ohrozit nemůže.

Celý program obsahuje všechny zmíněné procedury a přerušovací rutiny. Navíc je doplněn o inicializační funkce, které konfiguruje procesor, a o doplňkové procedury, které slouží pro generování zpoždění nebo pro blikání s diodami. Tyto procedury jsou využívány v hlavním programu. Vývojový diagram celého programu je zobrazen na obr. 14. Tento vývojový diagram odpovídá struktuře vykonávání programu v přístroji.



Obr. 14: Vývojový diagram finální verze programu pro procesor

1.3 FTDI Chip

1.3.1 Popis použitého obvodu

V konečném zapojení byl kvůli požadavku na připojení k USB sběrnici použit obvod FT232 RL od firmy FTDI Chip (www.FTDIChip.com). Tento obvod je podle [5] převodník ze sběrnice USB na obecný protokol UART včetně všech signálů pro hardwarové řízení toku a také plně podporuje softwarové řízení toku typu X-on/X-off. Obvod, kromě samotné konverze protokolů, nabízí i další možnosti, jako například vestavěný generátor hodinového signálu, který může být využit jako zdroj hodinového signálu pro další obvody, nebo přímé připojení led diod, zobrazující činnost vysílání nebo přijímání a dalších procesů. V moderních obvodech firmy FTDI je již integrována většina prvků potřebných pro připojení ke sběrnici USB, jako například zakončovací rezistory, výše zmíněný krystalový oscilátor nebo filtr napájecího napětí včetně resetovacího obvodu při připojení napájecího napětí, takže potřeba externích součástek je minimální. Obvody mají již plně integrovanou EEPROM paměť, která může být využita pro ukládání informací o konfiguraci programovatelných pinů, sériového čísla nebo popisu produktu, pokud jsou tyto možnosti využívány. Obvod je možné s převodníkem výstupních napětíových úrovní použít pro připojení na sběrnici RS232 nebo RS485, ale pro připojení k UART modulu procesoru není převodník potřeba, protože FT232RL je schopný na základě napájení vlastních výstupních obvodů dodávat různé výstupní logické úrovně, takže je možné ho použít v kombinaci s procesory s napájením 5 V i 3,3 V.

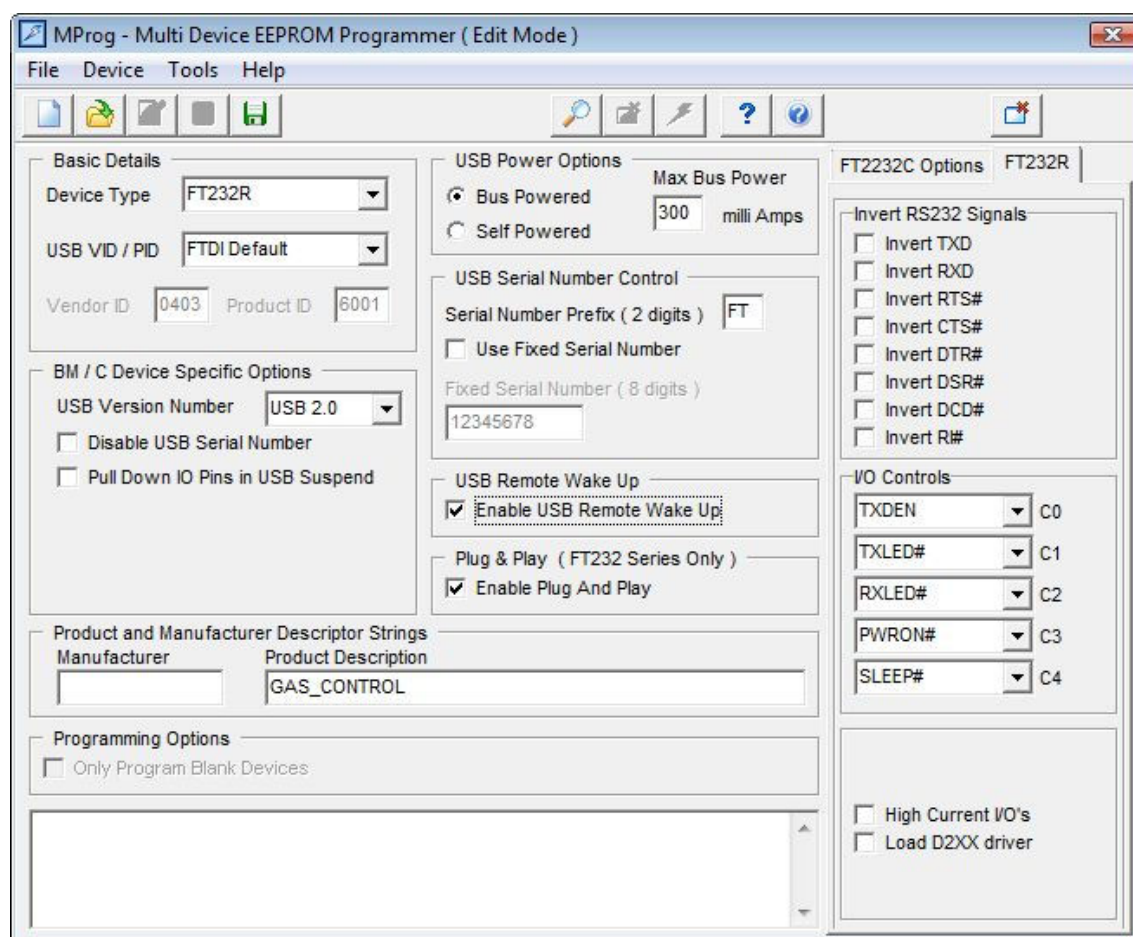
1.3.2 Konkrétní zapojení obvodu

V katalogovém listu [5] je několik typických zapojení s tímto druhem obvodu. Pro účely konstruovaného přístroje bylo zvoleno zapojení určené pro připojení k UART modulu procesoru, tudíž bez nutnosti konverze výstupních úrovní. Zapojení bylo modifikováno podle konkrétních potřeb aplikace. Napájení výstupních obvodů musí být na rozdíl od výše zmíněného zapojení v [5] napájeno napětím 3,3 V, protože mikroprocesor, ke kterému bude obvod připojen, pracuje s logickými obvody na úrovni 3,3 V. Pro generování této napětíové úrovně je obvod přizpůsoben, takže toto napětí může být odebíráno přímo z jeho vlastního pinu. Hlavní napájení obvodu (5 V) není

odebíráno z USB, ale z napájecího modulu (viz Napájecí modul). Pro potřeby aplikace není nutné zapojovat žádné programovatelné piny ani piny pro hardwarové řízení toku na sběrnici. Jediné signály připojené k procesoru tedy budou signály Rx a Tx. Konkrétní zapojení obvodu realizované v přístroji je zobrazeno ve schématu zapojení na obr. 19.

1.3.3 Programování FT232RL

Jak je výše uvedeno, tak obvod FT232RL má velké množství programovatelných vlastností, které se musí v obvodu nakonfigurovat. Protože je obvod primárně určen pro připojení ke sběrnici USB, tak i programování tohoto obvodu je realizováno přes tuto sběrnici. Výrobce (firma FTDI chip) k tomuto účelu dodává software MPROG, jehož pracovní prostředí je zobrazeno na obr. 15.



Obr. 15: Pracovní prostředí programu MPROG

Tímto programem je možné po sběrnici USB naprogramovat všechny obvody jejich výroby. Při konfiguraci je nutné nastavit minimálně základní parametry, jako typ použitého obvodu a použité napájení, a dále je zde možné nastavit většinu volitelných

možností, jako invertování výstupních pinů, nastavení programovatelných vstupů/výstupů a také popisný text k přístroji. Takto vytvořená konfigurace se následně zapíše do EEPROM paměti obvodu a tím je obvod připraven k používání.

1.4 Budič sběrnice CAN

Používaný procesor má, jak je výše uvedeno, integrovaný řadič sběrnice CAN. Z tohoto řadiče ovšem nelze přímo budit fyzickou vrstvu sběrnice. CAN je ve fyzické vrstvě implementovaná jako diferenciální sběrnice a procesor má na výstupech z modulu CAN (na pinech Rx a Tx) logické signály na úrovni 3,3 V. Pro konverzi do diferenciálních úrovní je použit obvod TLE 6250 od firmy Infineon (www.infineon.com). Obvod potřebuje pro funkci minimum externích součástek a je vyráběn ve verzích pro 5V i 3,3V systémy. Při použití verze pro 3,3V systémy lze obvod připojit přímo k procesoru a k napájení, jak ukazuje doporučené zapojení v [6]. Jak lze vidět v celkovém zapojení přístroje na obr. 19, tak obvod využívá pro funkci obě dvě napájecí napětí. Pro napájení koncového stupně na straně sběrnice CAN 5V úroveň a 3,3V úroveň pro napájení logických obvodů na straně procesoru. Kromě vstupních, výstupních a napájecích pinů má použitá verze obvodu ještě jeden logický vstup INH (inherited), kterým lze ovládat režim obvodu. Podle logické úrovně na tomto pinu lze obvod uvést do režimu standby (úsporný režim s odběrem řádově v μA) nebo do normálního režimu, kdy má obvod podle [6] poměrně značný odběr až 70 mA v dominantním stavu na sběrnici.

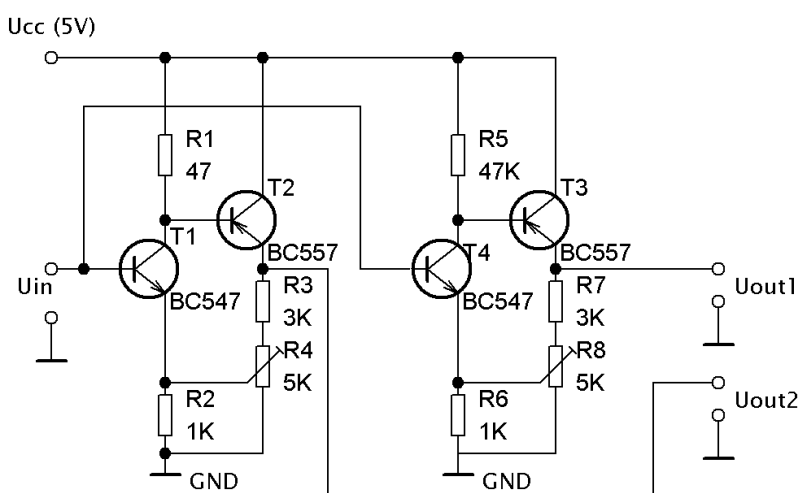
1.5 Koncový stupeň pro připojení k plynovému pedálu

Jak je zmíněno v teoretickém rozboru, tak napětí D/A převodníku na výstupu procesoru není dostatečné na přímé připojení místo plynového pedálu. Procesor je schopen generovat napětí až do výše použitého referenčního napětí, přičemž referenční napětí je povoleno připojit maximálně do výše napájecího napětí pro analogové periferie procesoru. Tato omezení determinují pracovní rozsah D/A převodníku na rozmezí maximálně 0-3,3 V. Toto rozmezí nepokryje celé pásmo využívané v pedálu, a proto je potřeba na výstup D/A převodníku zařadit koncový stupeň, který tato pásma přizpůsobí. Hlavní úlohou tohoto bloku bude zesílit napěťově výstupní signál a zároveň lze tento blok použít pro přesné generování dvojnásobného souběžného signálu (viz. 2.1). Pro každou variantu plynového pedálu bude muset být koncový stupeň jiný.

Z toho důvodu bude integrovaný do připojovacího kabelu k plynovému pedálu, který je taktéž pro každou variantu jiný. Koncový stupeň bude realizován jako dvojitý napěťový zesilovač pro varianty pedálu se dvěma napětími a pro varianty s jedním signálem pouze jako zesilovač jednoduchý.

Při realizaci napěťového zesilovače je nutné dodržet několik zásad, které jasně určují, jak daný zesilovač sestavit. Základním požadavkem je jeho pracovní rozsah. Zesilovač musí pracovat od nuly do hodnoty napájecího napětí. Dalším požadavkem je konstantní zesílení v pracovním pásmu. Základní myšlenkou při realizaci koncového stupně bylo využít zapojení operačního zesilovače jako neinvertujícího zesilovače. Toto zapojení je velice jednoduché, ale aby mohl operační zesilovač pracovat od nulové hodnoty napětí na výstupu, potřebuje symetrické napájecí napětí, které v obvodu není. Existují sice i zapojení operačních zesilovačů bez symetrického napájení, ale počet součástek nutných k realizaci značně vzroste. Takové zapojení ztrácí největší výhodu, což je jeho jednoduchost. V konečné a vyzkoušené verzi koncového stupně jsou využity dva tranzistorové zesilovací stupně, každý složený ze dvou tranzistorů. S minimálním počtem součástek je toto zapojení na obr. 16 plně vyhovující výše popsaným požadavkům a navíc relativně jednoduché. Do zapojení byl umístěn odporový trimr ve zpětné vazbě, který slouží pro přesné nastavení souběžného výstupního napětí pro kalibraci. Zesílení obou stupňů je dáno vztahy (5 a 6).

$$A_{U1} = \frac{R2 + R3 + R4}{R2} \quad A_{U2} = \frac{R6 + R7 + R8}{R6} \quad (5, 6)$$



Obr. 16: Zapojení koncového stupně pro připojení k plynovému pedálu

Pro základní přizpůsobení signálu na úroveň plynového pedálu je nutné dosáhnout zesílení s hodnotou 2. Pro variantu se dvěma signály je potřeba druhý signál přizpůsobit zesilovačem se zesílením 4. Celkové zapojení jednoho z koncových stupňů pro variantu se dvěma signály je na obr. 16. Hodnoty rezistorů v obou stupních jsou zvoleny tak, aby bylo možné dosáhnout obou výše zmíněných hodnot zesílení. Hodnoty zesílení se přesně nastavují trimrem při kalibraci.

1.6 Napájecí modul

Navrhovaný přístroj bude později sloužit, jak je dáno v zadání, k řízení automobilu při akustických testech. Pro tento účel je tedy krajně nevhodné, aby byl napájen ze síťového napětí. Pro napájení se nabízí několik variant, které je potřeba zvážit a zpracovat s ohledem na potřeby aplikace, co se týče stability napájecího napětí a velikosti proudového odběru. V přístroji je kromě procesoru samotného několik obvodů respektive funkčních bloků, které je potřeba napájet různým stabilizovaným napětím.

První variantou zdroje napájení je sběrnice USB. Z konektoru USB lze odebírat již stabilizované napětí 5V, které je ovšem limitováno proudovým odběrem. Při nastavení FTDI obvodu (viz výše), který slouží v přístroji, jako rozhraní se sběrnicí USB je jeden z důležitých parametrů předpokládaný proudový odběr. Při překročení této nastavené hodnoty může totiž podle [7] dojít k samovolnému odpojení obvodu od USB, čímž počítač chrání vlastní obvody proti přetížení. Obzvláště náchylné na tyto problémy jsou notebooky, ke kterým bude přístroj připojován pravděpodobně nejvíce. Hazardní stavy, které by nastaly při výpadku napájení za běhu programu, by vedly k potížím s řídicí jednotkou motoru. Ta by se domnívala, že došlo k poruše plynového pedálu a vyhodnotila by tento stav jako poruchu (viz 2.1). Z těchto důvodů je napájení z konektoru USB nevhodné. Další možností je využít napětí v diagnostické zásuvce. V této zásuvce je k dispozici stabilizovaných 12V z palubní sítě automobilu. Zařízení bude diagnostickou zásuvku využívat pro čtení dat ze sběrnice CAN, a proto je připojení k tomuto konektoru nezbytné. Využít tento konektor zároveň jako zdroj napájení se jeví jako ideální. Jediná nevýhoda, kterou toto řešení má, je, že při programování přístroje ze sběrnice USB by přístroj musel být připojen k automobilu. To prakticky zcela vylučuje použití přístroje s pevným počítačem. I když je pro programování přístroje počítáno hlavně s notebooky, je dobré myslet na universálnost návrhu, a proto umožnit napájet přístroj i jinak než z automobilu. Pro režim

naprogramování by bylo možné použít akumulátor, který by byl vestavěný v přístroji a dobíjel by se při připojení k automobilu, ale protože bude přístroj při programování vždy určitě připojen k USB, je použití akumulátoru zbytečně složité a lepší variantou je zde využít napětí na sběrnici USB. Toto napětí je sice zatíženo výše zmíněnými problémy, ale při programování přístroje bude proudový odběr omezený, protože je budič sběrnice CAN uveden do úsporného režimu. Návrh stabilizovaného zdroje tedy vychází z několika následujících předpokladů. Napájecí napětí na vstupu zdroje bude možné dodávat z diagnostické sběrnice v automobilu nebo ze sběrnice USB, přičemž prioritu bude mít napětí z diagnostické sběrnice z důvodů, které budou dále vysvětleny. Na výstupu zdroje potřebuje přístroj dvě hodnoty stabilizovaného napětí. Pro napájení procesoru, diod a tlačítek je potřeba 3,3 V a pro napájení obvodu FTDI, koncového stupně pro plynový pedál a koncového obvodu pro sběrnici CAN je potřeba 5 V. Budič sběrnice CAN využívá obě napájecí napětí.

Obě vstupní napětí zdroje jsou již stabilizovaná, ale při napájení z diagnostické zásuvky v automobilu je na vstupu 12 V. Vstupní napětí musí být přizpůsobeno na dvě hodnoty napájecího napětí (viz výše). K tomuto účelu byl vybrán stabilizátor TLE4476 od firmy Infineon (www.infineon.cz). Tento obvod v jednom pouzdře integruje dva nízkoubytkové stabilizátory napětí. Na jeho výstupních svorkách lze odebírat obě potřebná napětí pro přístroj. Maximální povolené proudové odběry jsou podle [8] 430 mA z 5V větve a 380 mA z 3,3V větve obvodu. Řešení s tímto obvodem má jeden nedostatek, který ovšem na funkci přístroje nemá vliv. Stabilizátor, aby na výstupu zabezpečil napětí požadovaných parametrů, potřebuje ke své vlastní činnosti energii. I když je používán obvod tzv. „low-drop“ stabilizátor (s nízkým úbytkem napětí na součástce), tak při maximálním vytížení má podle [8] typický úbytek napětí na vnitřních obvodech 0,3 V. Tento fakt se projeví, při napájení zdroje ze sběrnice USB, i když ne v kritických hodnotách, protože proudové odběry ze zdroje nejsou na hranici maximálních povolených hodnot (viz tabulka na obr. 17).

| Spotřebič | Odběr z 5V větve [mA] | Odběr z 3,3V větve [mA] |
|----------------|-----------------------|-------------------------|
| C8051 F040 | - | 17 + 30 (PORTY) |
| FT 232R | 15 | 0 |
| TLE 6250 | 70 | 70 |
| Koncové obvody | 5 CAN + 10 Pedál | - |
| Celkem | 100 | 117 |

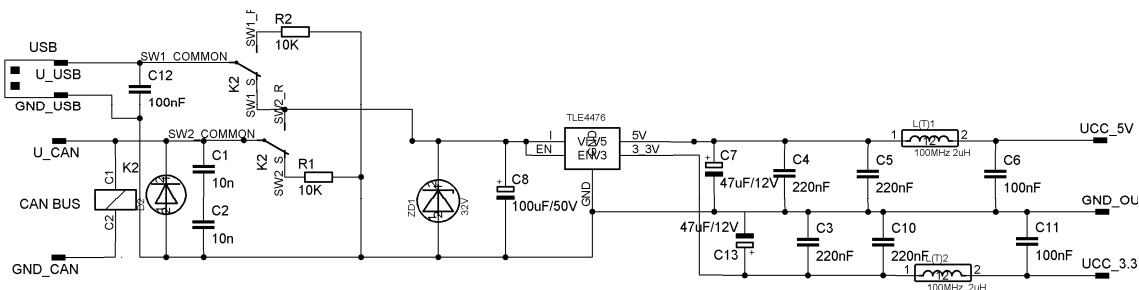
Obr. 17: Tabulka maximálních proudových odběrů ze zdroje

Vstupní napětí pro zdroj je 5 V a při nejhorších bilancích by mohlo dojít ke stavu, že vyšší napájecí napětí dosáhne pouze hodnoty 4,7 V. Na funkci přístroje to ovšem vliv mít nebude, protože při programování přístroje (napájení z USB) není potřeba přesné napájecí napětí pro koncový stupeň pedálu ani komunikace na sběrnici CAN. Obvod FTDI, který je jako jediný v tomto režimu potřeba, je podle [5] k takovéto hodnotě tolerantní. Při připojení k automobilu dojde k přepnutí na napájení z diagnostické zásuvky (napětí na vstupu zdroje je 12 V), takže obvod bude mít dostatečně velké napětí na vstupu a výstupní napětí budou podle [8] obě v tolerancích $\pm 4\%$. Přepínání mezi zdroji vstupního napětí je v přístroji realizováno pomocí relé. Toto řešení má sice všechny nevýhody kontaktního spínání, ale zabezpečuje galvanické oddělení obou napájecích napětí v případě, že by byla připojena najednou. V takovém případě je kvůli výše popsané skutečnosti zdroj napájen 12 V z diagnostické sběrnice. Na výstupu zdroje je sada filtračních a vyhlazovacích kondenzátorů, které vychází z požadavků na kvalitu napájecího napětí pro všechny prvky v přístroji. Celkové zapojení zdroje napájecích napětí je ve schématu na obr. 18.

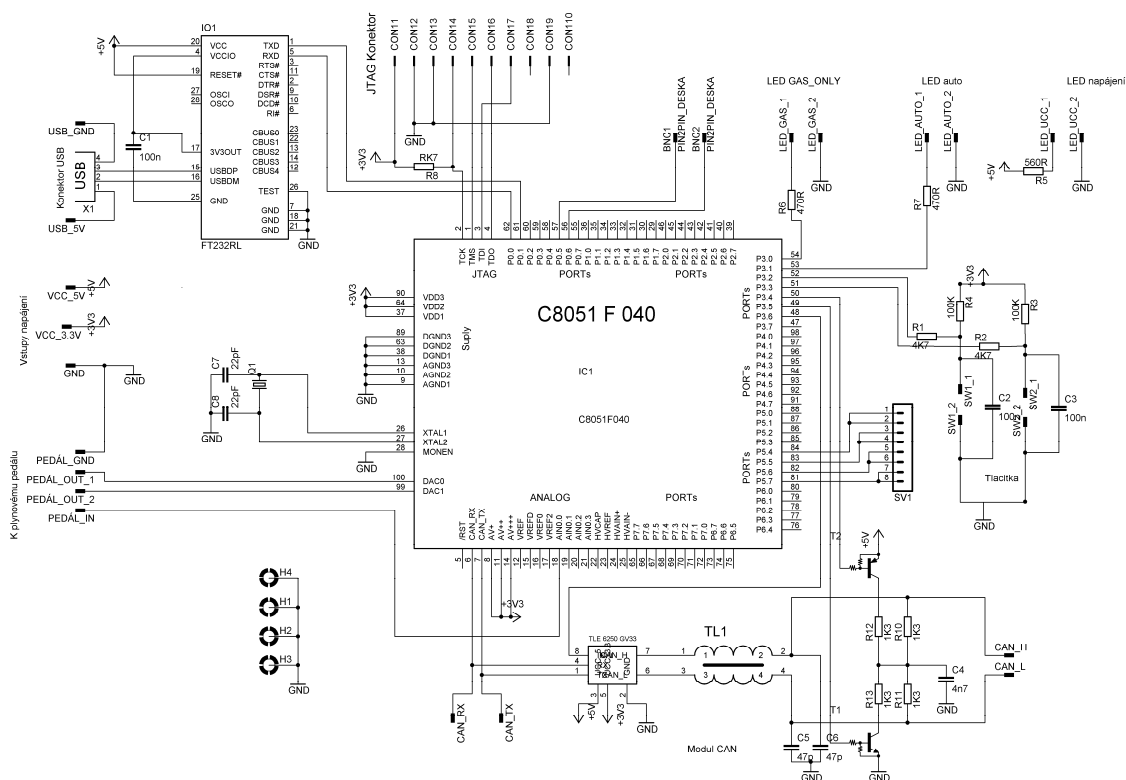
1.7 Plošné spoje

Pro konečnou podobu přístroje je zapotřebí navrhnout desky plošných spojů, které budou zasazeny do krabičky. Celý přístroj je rozdělen do dvou částí. Desky zdrojové a signálové části jsou odděleny z několika důvodů. Při oddělení zdrojové a signálové části do samostatných desek je umožněno samostatně testovat oba celky a lépe tím diagnostikovat případné závady, které nastávají při ožívování obvodů. Dalším důvodem je návrhový software. Pro návrh desek plošných spojů byl použit program EAGLE (www.elcad.cz/eagle), konkrétně volně šiřitelná verze tohoto softwaru. Tato verze je funkční, ale omezená následujícími parametry. Při návrhu lze aplikovat spoj pouze do dvou vrstev a schéma lze kreslit pouze na jeden list. Tyto parametry nijak neomezuji návrh konkrétních desek, neboť se dvěma vrstvami bylo počítáno od začátku návrhu a stejně tak i schéma lze bez problému nakreslit do jednoho listu. Jediným parametrem, který limituje návrh, je maximální velikost navrhované desky 100×80 mm. Při návrhu zdrojové i signálové části do tohoto rozměru by bylo velmi obtížné dodržet návrhová doporučení pro tvorbu desek plošných spojů. Také z tohoto důvodu byl přístroj rozdělen do dvou desek.

Většina součástek pro obě části byla volena v provedení SMD, aby bylo docíleno menších rozměrů finálního výrobku. Zapojení jednotlivých prvků v obou částech jsou na obr. 18 a obr. 19. Jejich podoba vychází z výše popsanych požadavků na obvody.



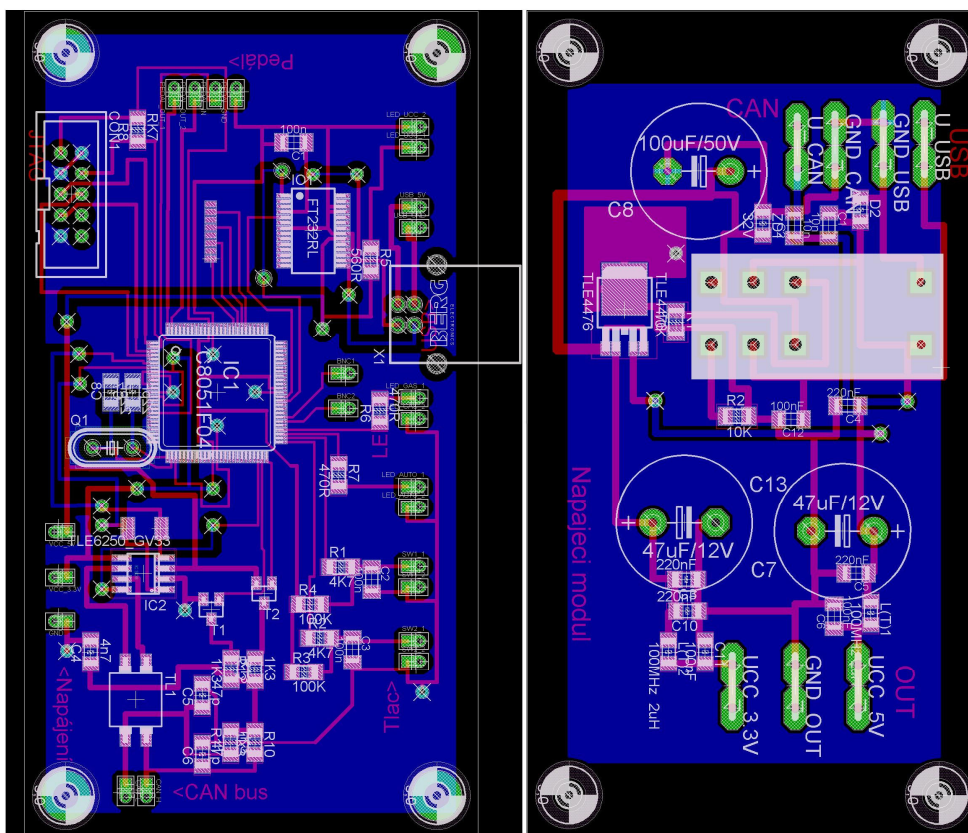
Obr. 18: Schéma zapojení přístrojového zdroje



Obr. 19: Schéma zapojení signálové části s procesorem

Do obvodu signálové části byl přidán běžně užívaný přístrojový konektor USB typu B pro připojení k PC. Ostatní vstupy a výstupy i propojení mezi oběma deskami je realizováno pomocí univerzálních odlamovacích konektorů tak, aby mohly desky být nezávisle vyměněny nebo nahrazeny případnou novější verzí. Všechny ovládací a zobrazovací prvky, jako tlačítka a diody jsou také vyvedeny na konektory, protože budou umístěny ve víku krabíčky. V konečné podobě přístroje budou oba plošné spoje položené nad sebou na distančních sloupcích a zapracovány do krabíčky. Rozměry obou

desek a rozteč děr pro montáž jsou proto totožné. Výsledné desky plošných spojů jsou zobrazeny v měřítku 1:1 na obr. 20.



Obr. 20: Motivy desek plošných spojů včetně součástek

2 Ovládací aplikace

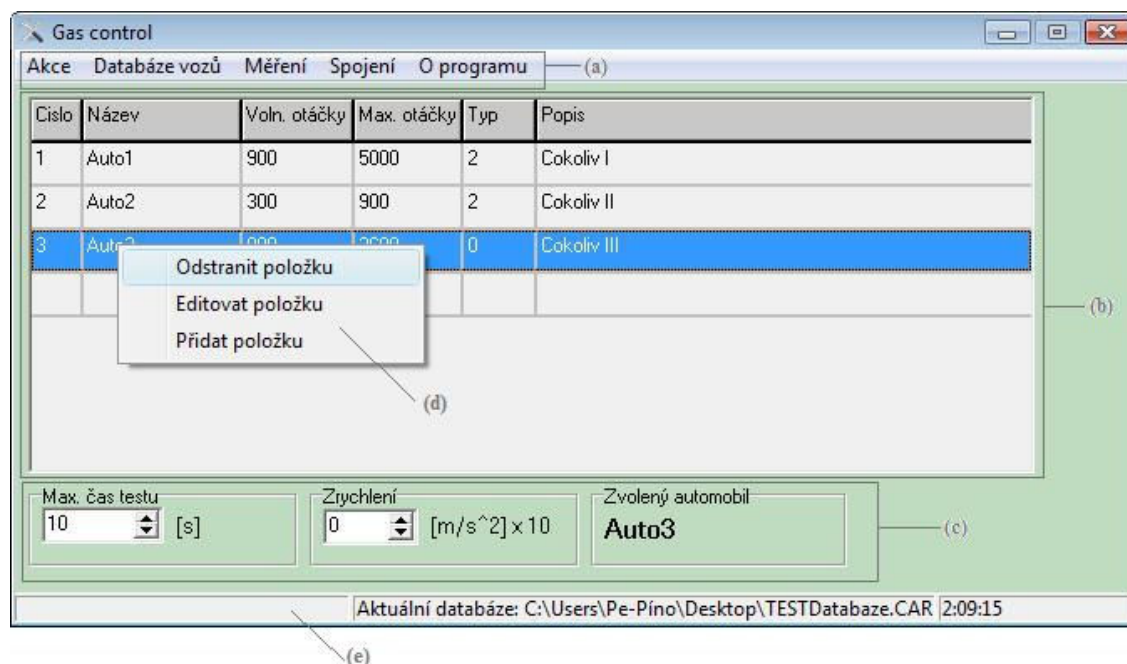
2.1 Základní princip a požadavky

Přístroj pracuje se zadaným programem, který je potřeba do něj uložit. K tomuto účelu je určena počítačová aplikace. Mezi hlavní funkce aplikace patří vedení databáze automobilů. Tuto databázi je možné v aplikaci editovat. Výběrem položky v graficky zobrazené databázi je zvolen automobil určený k testu. Zvolením ostatních parametrů testu je plně definovaná testovací sada, kterou lze následně z aplikace odeslat do přístroje. Data jsou v aplikaci zpracována tak, aby procesor v přístroji data pouze uložil a používal. Případné převádění nebo upravování dat by procesor zbytečně zpomalovalo. Přístroj bude hardwarově připojen k počítači přes sběrnici USB. Obvod FT232 RL, který je v přístroji použit jako rozhraní s počítačem, umožňuje použít USB z pohledu programování skrýt respektive převést na komunikaci po standardní sériové lince. Na straně PC se pak pracuje se sériovým portem, který má větší přenosové rychlosti než standardní sériový port. Aplikace je navržena pro spouštění pod operačními systémy Windows XP a vytvořena v prostředí Delphi7. Pro obsluhu sériového portu jsou využívány funkce Windows API. Volně dostupné jsou i knihovny pro obsluhu sériového portu, ale použitím API funkcí, které jsou navrženy speciálně pro operační systémy Windows, se program stane robustnější.

Databáze vedená v aplikaci je jednoduchá a proto využívá pouze ukládání do textového souboru typu (*.CAR). Soubor s touto příponou je navržen pro tyto účely a je vytvářen přímo v aplikaci. Formát souboru je obdobný jako u běžně používaných souborů typu (*.csv) (jednotlivé položky oddělené středníkem). Kromě editace přímo v aplikaci lze soubor otevřít například v textových programech. Aplikace pracuje pouze se dvěma typy souborů a jiné soubory neumožňuje filtr otevřít. Jedním ze souborů je již popsáný soubor databáze a dalším možným výstupem respektive vstupem aplikace je soubor (*.MEAS). Tento soubor má obdobný formát jako soubory databáze. Při ukládání do tohoto souboru se ovšem neukládá celá databáze, ale pouze aktuálně zvolený automobil a navíc nastavené hodnoty parametrů testu. Tato funkce umožňuje uživateli vytvořit si sadu často využívaných měření, pro které není nutné nastavovat všechny parametry.

2.2 Pracovní prostředí

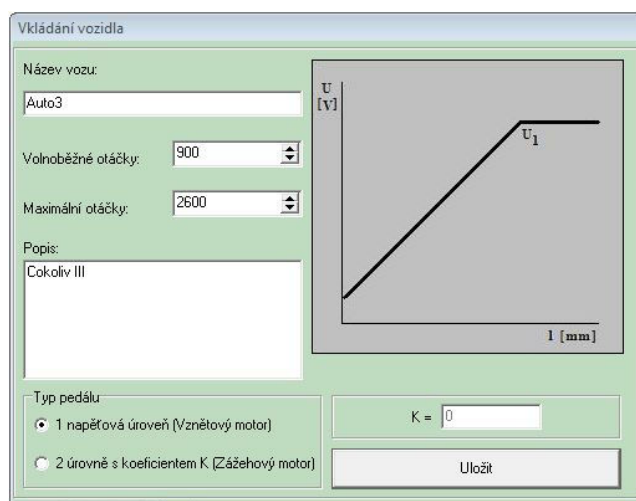
Pracovní prostředí aplikace se skládá ze tří základních oken. Hlavním požadavkem na celou aplikaci je intuitivní a přehledné ovládání. Hlavní okno programu, zobrazené na obr. 21, poskytuje všechny funkce programu v horním roletovém menu (a), které je běžné pro aplikace pod operačními systémy Windows.



Obr. 21: Hlavní okno ovládací aplikace Gas control

V hlavní tabulce (b) je zobrazena aktuálně otevřená databáze. Kliknutím na položku v tabulce se položka označí jako aktuálně zvolená a její název se zobrazí mezi parametry testu. Všechny další parametry testu se nastavují v oblasti (c) pod tabulkou. Kliknutím pravým tlačítkem do tabulky se zobrazí lokální roletové menu (d). Položky v tomto menu jsou různé podle aktuálního stavu databáze. Není-li databáze načtená, zobrazí se v tomto menu pouze tlačítko pro načtení databáze. Při načtené databázi jsou v tomto menu tlačítka pro editaci, přidání nebo odebrání položky v databázi (viz obr. 21). Informace o aktuálním stavu programu, jako například aktuálně otevřená databáze nebo připojení k portu, se zobrazují v dolní stavové liště (e), která je opět běžná u většiny aplikací.

Při přidávání vozidla do databáze se používá druhé okno programu. Toto okno je pouze dialogové a slouží k nastavení všech parametrů nově vkládaného automobilu (obr. 22). Pro nastavení parametrů slouží různé ovládací prvky formuláře. Pro přehlednost je při zadání typu plynového pedálu zobrazen napěťový průběh s vyznačenými veličinami. Tlačítkem „Uložit“ se data z formuláře uloží do databáze a formulář se zavře. Uživatel dále pokračuje v hlavním okně programu.



Obr. 22: Dialogové okno pro přidání automobilu

Třetí okno programu je používáno pro zadávání parametrů komunikačního portu. Toto okno (obr. 23) je opět dialogové a není možné ho minimalizovat ani maximalizovat. Ve dvou rozbalovacích nabídkách uživatel vybere číslo portu, ke kterému je přístroj připojen, a zvolí komunikační rychlost, která je v aktuální verzi přístroje nastavena na 115200 kb/s. Tlačítkem „Nastavit“ se formulář nastaví, ale program k portu ještě připojen není. Připojení je přístupné v hlavním okně z hlavního menu. Pokud připojení proběhne správně, ve spodní stavové liště programu se zobrazí informace, k jakému portu je přístroj připojen. Pokud připojení nebo jiná akce vyvolá chybu, program ji nahlásí zprávou. Zprávami program oznamuje i důležité informace v průběhu používání programu.

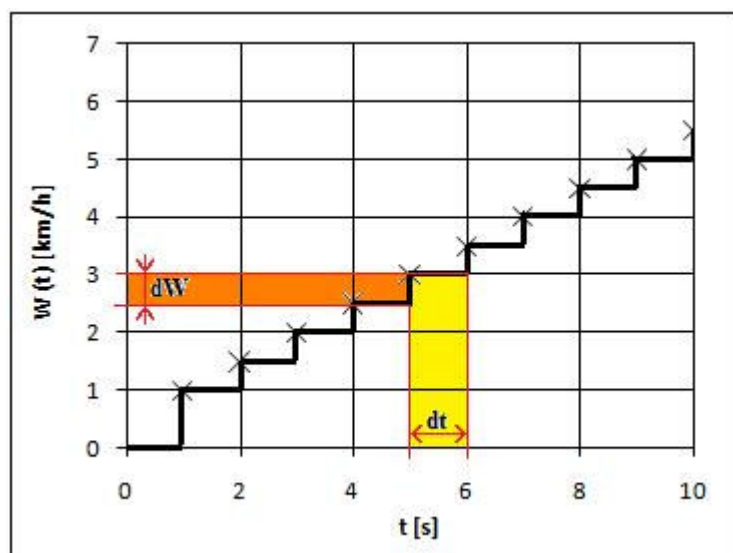


Obr. 23: Dialogové okno pro zadávání parametrů připojení

Bližší popis jednotlivých komponent a jejich používání za běhu programu je uveden v manuálu přístroje (viz 3).

2.3 Komunikační protokol

Po nastavení všech parametrů testu může uživatel test uložit nebo ho přímo nahrát do připojeného přístroje. Jak bylo výše uvedeno, tento přenos je realizovaný reálně po sběrnici USB ovšem programově po klasické sériové sběrnici. Aby data vyslaná z aplikace mohl přístroj respektive procesor správně reprezentovat, musí být na obou stranách komunikačního kanálu implementovaný stejný komunikační protokol. Při přenosu se jedná o malý balík dat o velikosti několik bajtů, a proto je komunikace realizována bez řízení toku na sběrnici a protokol je navržený hlavně s důrazem na jednoduchost a minimalizaci objemu přenášených dat. Základní myšlenkou při navrhování protokolu bylo vyslat do přístroje datovou řadu, která by reprezentovala žádané hodnoty otáček nebo rychlosti v předem určených časových rozestupech. Takovýto protokol by byl absolutně univerzální co do žádaného průběhu zmíněných veličin, ale pro konkrétní zadání je zbytečný, protože je vyžadováno pouze řízení lineárního průběhu rychlosti, respektive konstantního zrychlení. Po analýze možností byl navržen protokol, který v průběhu testování aplikaci vyhovuje. Datová řada byla nahrazena dvěma hodnotami, které stačí k plnému popsání lineární závislosti. První hodnotou je hodnota inkrementu a druhá času inkrementu. Tyto dvě hodnoty popisují element diskrétního lineárního průběhu, jak je zobrazeno na obr. 24. Hodnota inkrementu v datovém protokolu znázorňuje hodnotu elementu dW a hodnota času inkrementu znázorňuje hodnotu elementu času dt . Dalšími daty, která je potřeba přenést do přístroje, jsou hodnoty maximálních a volnoběžných otáček testovaného automobilu. Na základě těchto parametrů může přístroj kontrolovat a nastavovat otáčky motoru tak, aby nedošlo k zastavení motoru ve volnoběhu nebo nedošlo k přesáhnutí maximálních otáček motoru. Posledním částí protokolu jsou informační data, která v současné verzi nesou pouze informace o druhu plynového pedálu v řízeném automobilu.



Obr. 24: Diskrétní průběh žádané hodnoty (W)

Data přenášená do procesoru se v aplikaci přizpůsobují pro snadné zacházení procesoru s nimi. Dva údaje o otáčkách a inkrement rychlosti jsou přizpůsobeny podle dat na straně procesoru. Procesor aktuální hodnotu těchto veličin získává z diagnostické sběrnice a proto je tyto data potřeba přizpůsobit těmto formátům. Hodnota času inkrementu je pro procesor převedena do milisekund. Na straně procesoru je totiž nastaven časovač, který poskytuje časový údaj na 1 ms, a proto se tomuto formátu přizpůsobují přenášená data. Typ automobilu je zakódovaný tak, že při plynovém pedálu s jedním napětím se přenáší nula. Při variantě pedálu se dvěma napěťovými signály je hodnota této informace shodná s koeficientem druhého napětí. Výsledný datový protokol, používaný při přenosu dat do přístroje je zobrazený (včetně datových typů používaných v procesoru) v tabulce na obr. 25. Jak jde vidět, tak celý přenos je realizován pomocí 8 datových bajtů, což je významná úspora systémových prostředků procesoru při přijímání a také času přenosu oproti původnímu záměru popsanému výše.

| Typ pedálu | Inkrement | Čas inkrementu | Max. otáčky | Voln. otáčky |
|-------------|--------------|----------------|--------------|--------------|
| 8 b - uchar | 16 b - short | 8 b - uchar | 16 b – short | 16 b – short |

Obr. 25: Formát přenosového protokolu

Kompletně okomentovaný zdrojový kód aplikace je přiložen na CD.

3 Manuál přístroje

Manuál přístroje byl zhotoven jako samostatná publikace, která je určena zejména pro obsluhu přístroje. V následujících kapitolách je kompletní podoba manuálu bez čelní strany a obsahu. Některé pasáže popisující aplikaci jsou proto velmi podobné kapitole 2.2.

3.1 Úvod

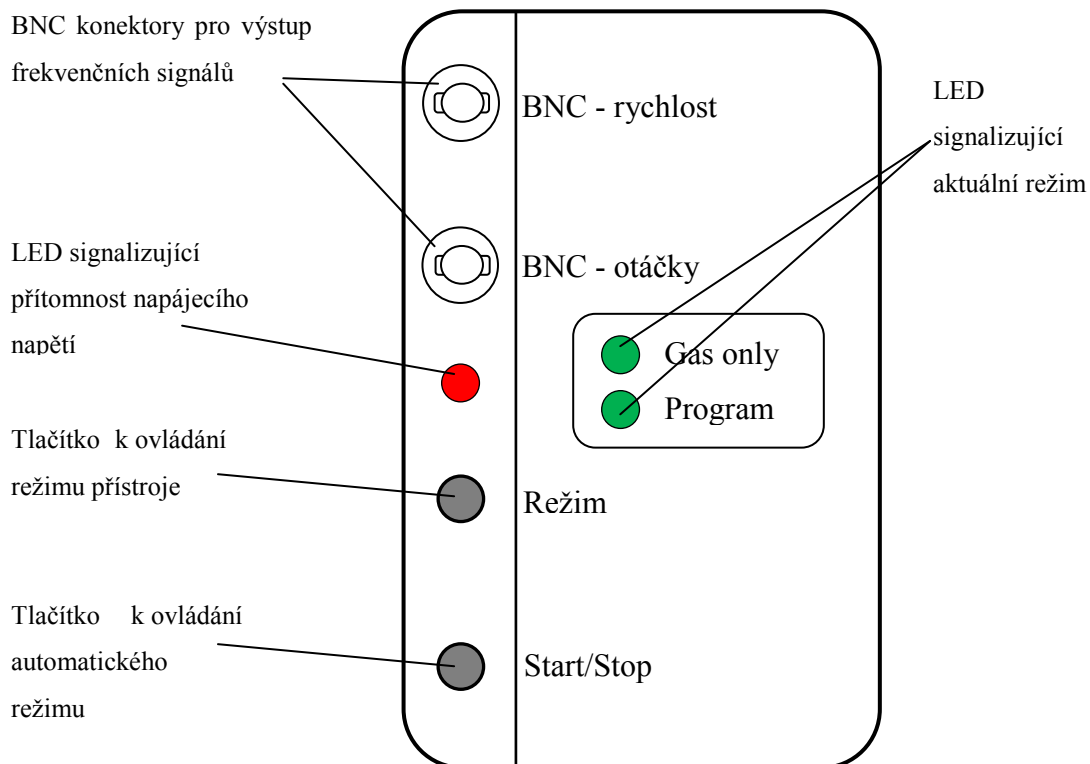
Tato publikace slouží jako manuál přístroje Gas control s popisem a návodem na obsluhu počítačové aplikace určené k ovládání přístroje i přístroje samotného.

3.2 Přístroj Gas control

- slouží k řízení rychlosti automobilu podle nadefinovaného průběhu
- převádí otáčky motoru a rychlost automobilu ze sběrnice CAN na frekvenci
- je určený pro používání ve všech současných vozech Škoda auto
- k otevření komunikace na sběrnici CAN využívá přístroj COTR 03 firmy EHL elektronika
- pro připojení k plynovému pedálu má několik variant připojovacích kabelů, podle typu pedálu v automobilu (viz 3.2.2)
- je programovatelný prostřednictvím aplikace Gas control
- k PC se připojuje přes port USB

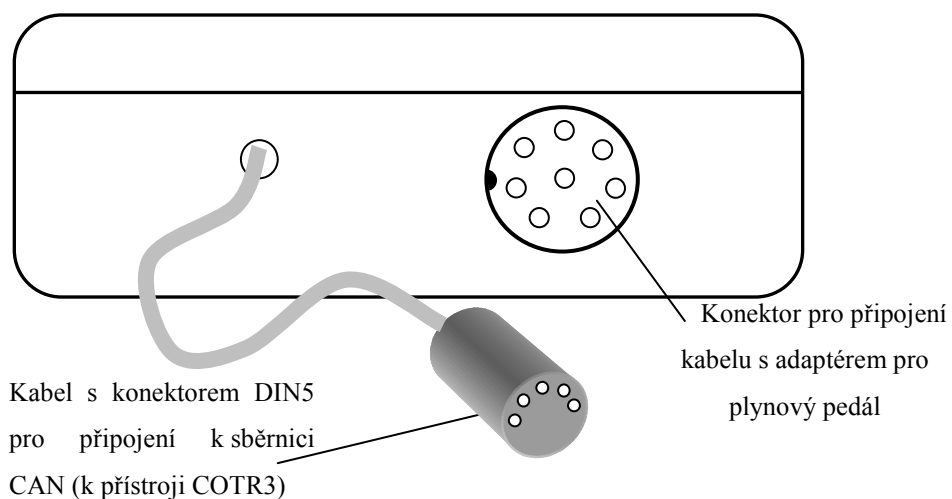
3.2.1 Rozložení prvků na přístroji

Čelní panel

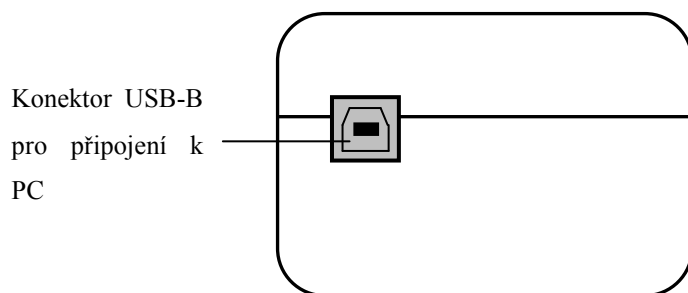


Obr. 26: Pohled na prvky přístroje (pohled shora)

Boční strany



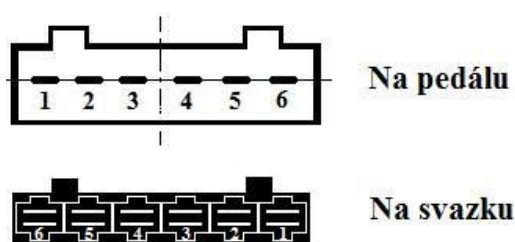
Obr. 27: Pohled na prvky přístroje (pohled zleva)



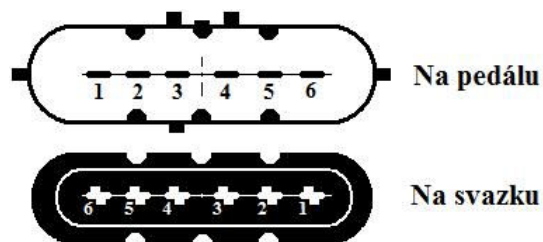
Obr. 28: Pohled na prvky přístroje (pohled zezadu)

3.2.2 Kabely pro připojení k pedálu

V současných modelech automobilů jsou zastoupeny tři varianty plynových pedálů. Pro správnou funkci přístroje je zapotřebí použít správný kabel pro připojení. U starších vozů (Fabia, Octavia) jsou používány pedály s konektory zobrazenými na obr. 29. U novějších vozů je konektor nahrazen bezpečnějším, který je vyobrazen na obr. 30. Od těchto základních dvou typů se odvíjí dva druhy připojovacích kabelů.



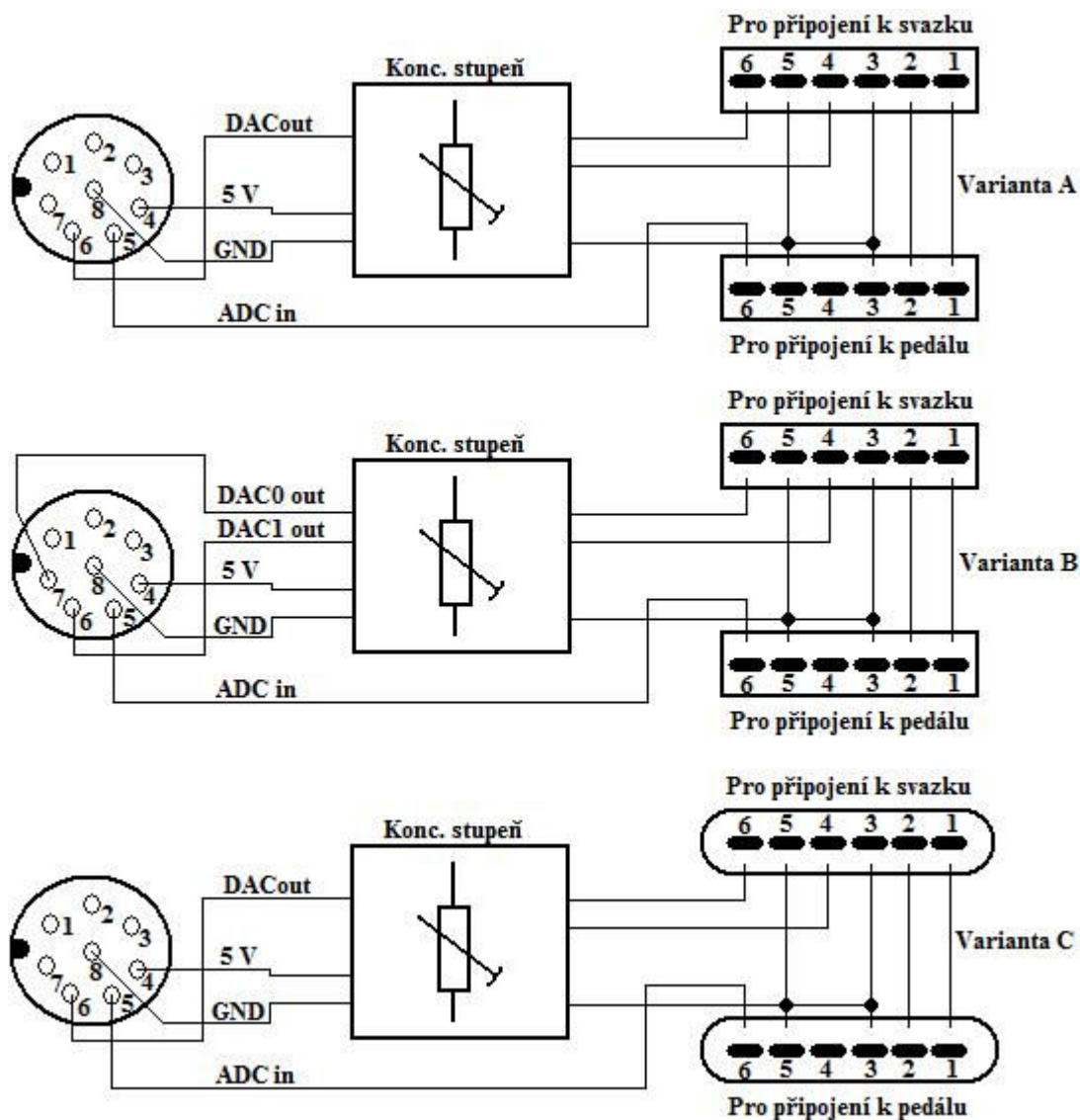
Obr. 29: Staré konektory pedálu



Obr. 30: Nové konektory pedálu

Třetí varianta připojovacího kabelu je modifikací kabelu pro starší verze. Konektory jsou stejné, jen vodiče a koncový stupeň jsou zapojeny jinak. Tato modifikace je nutná, protože pro dieselové motory nejsou v plynovém pedálu dvě napěťové úrovně, ale pouze jedna, doplněná diskretní hodnotou ze spínače Kick-down. Jak již výše zmíněno, každý kabel pro připojení k plynovému pedálu má zabudovaný koncový stupeň, který upravuje signál z přístroje na signál, který bude bezpečně vyhodnocovat řídicí jednotka motoru. Tyto koncové stupně jsou vybaveny odporovým trimrem pro kalibraci zesílení koncového stupně. Toto nastavení je důležité kvůli souběhu napětí, které kontroluje řídicí jednotka motoru. Výsledkem jsou tři varianty kabelů, které jsou schematicky znázorněny na obr. 31. Varianta A odpovídá vozům se starým konektorem a zážehovým motorem. Varianta B vozům se starým konektorem, ale vznětovým motorem a

Varianta C odpovídá vozům s novými konektory.



Obr. 31: Varianty zapojení kabelů pro připojení k plynovému pedálu

3.2.3 Režimy přístroje

Přístroj pracuje ve dvou základních režimech, v režimu „Gas only“ a v programovém režimu. Třetí režim určený pro kalibraci koncových členů je využíván jen při kalibraci viz (3.4.4). Po připojení přístroje k napájení, ať už konektorem USB nebo konektorem DIN5, je přístroj uveden do režimu „Gas only“.

Funkce jednotlivých režimů:

„Gas only“: Tento výchozí režim slouží pro nastartování automobilu a pro případné přejíždění s automobilem při připojení přístroje. V tomto režimu se přístroj chová transparentně a pouze kopíruje do řídicí jednotky motoru napětí úrovně z plynového pedálu. Vozidlo je absolutně pod kontrolou běžným plynovým pedálem.

Programový režim: V tomto režimu nemá plynový pedál vliv na chování automobilu. Po uvedení přístroje do tohoto režimu dojde k načtení programu z paměti FLASH. Pokud přístroj ještě není naprogramovaný, nebo je-li program ve Flash paměti přístroje poškozený, musí být přístroj znovu naprogramován prostřednictvím PC, a proto hlásí chybu (viz 3.2.5). Po načtení programu začne přístroj regulovat otáčky motoru na hodnotu volnoběžných otáček². Po spuštění chodu programu, který je signalizován blikáním LED diody „Program“, začne přístroj regulovat rychlost automobilu na hodnotu danou časovým průběhem rychlosti, který byl do přístroje naprogramován. Po dosažení maximálních otáček³ dojde k ukončení chodu programu a přístroj reguluje otáčky automobilu opět na volnoběžné otáčky. Program lze znovu spustit nebo přepnout přístroj do režimu „Gas only“.

Kalibrační režim: Přístroj generuje na výstupní piny plynového konektoru konstantní napětí, které slouží pro kalibraci.

3.2.4 Popis ovládacích prvků

Tlačítko „Režim“: Stisknutí vyvolá změnu režimu přístroje. Aktivní je vždy, i při chodu programu. Při stisknutí v aktivním programovém režimu se vykonávání programu ukončí a přístroj přejde do režimu „Gas only“. Dlouhým stisknutím tohoto tlačítka (cca 5 sekund) uvedete přístroj do Kalibračního režimu. Z tohoto režimu se lze vrátit pouze stisknutím tlačítka Start/Stop.

Tlačítko „Start/Stop“: Stisknutí vyvolá spuštění automatického chodu v programovém režimu. Toto tlačítko je aktivní pouze v programovém režimu. Stisknutí tlačítka v režimu „Gas only“ nemá na přístroj vliv.

3.2.5 Popis signalizačních prvků

Červená LED: Signalizuje napájecí napětí v digitální části obvodu.

Zelená LED „GAS only“: Signalizuje aktuálně zvolený režim „Gas only“.

² Volnoběžné otáčky nastavené v PC aplikaci při programování přístroje, nikoli tovární nastavení volnoběhu automobilu!

³ Maximální otáčky nastavené v PC aplikaci při programování přístroje, nikoli tovární nastavení omezovače automobilu!

Zelená LED „Program“:

Svítí: Signalizuje aktuálně zvolený programový režim.

Bliká: Signalizuje spuštěný automatický chod v programovém režimu.

Při **střídavém blikání zelených diod** zobrazujících režim, záleží na předchozím vykonaném kroku:

Po naprogramování přístroje z PC: Blikání signalizuje úspěšně přijatý balík programovacích dat a jejich uložení do Flash paměti. Přístroj po krátké době samovolně přejde do režimu „Gas only“.

Po spuštění automatického chodu v programovém režimu: Došlo k chybě při načtení programu z FLASH paměti přístroje. Přístroj po krátké době samovolně přejde do režimu „Gas only“.

Při **blikání obou diod současně** je přístroj v kalibračním režimu.

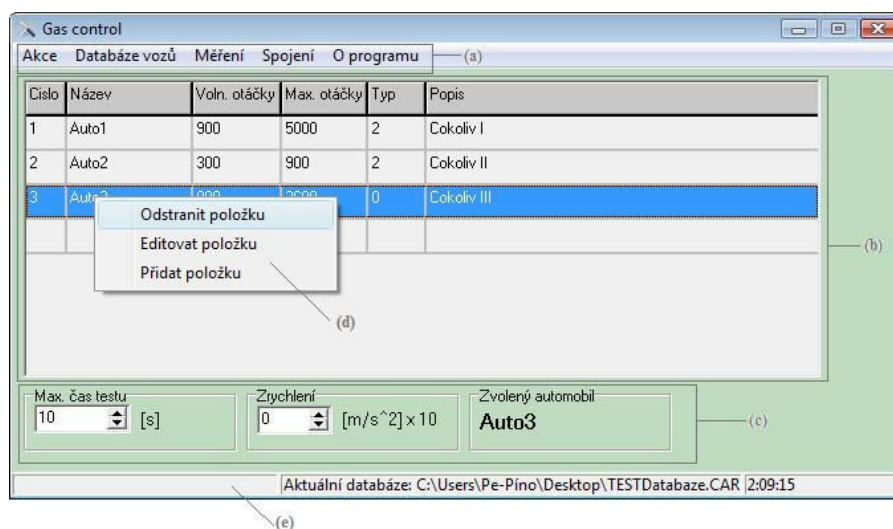
3.3 Počítačová aplikace Gas control

Aplikace Gas control slouží k naprogramování přístroje Gas control. V aplikaci je vedena databáze automobilů, kterou si uživatel vytváří vkládáním jednotlivých automobilů. V databázi jsou uloženy všechny potřebné technické parametry jednotlivých vozů. Při připojení přístroje je možné ho naprogramovat z pracovního prostředí aplikace.

3.3.1 Popis pracovního prostředí

Pracovní prostředí aplikace se skládá ze tří základních oken.

Hlavní okno



Obr. 32: Hlavní okno ovládací aplikace Gas control

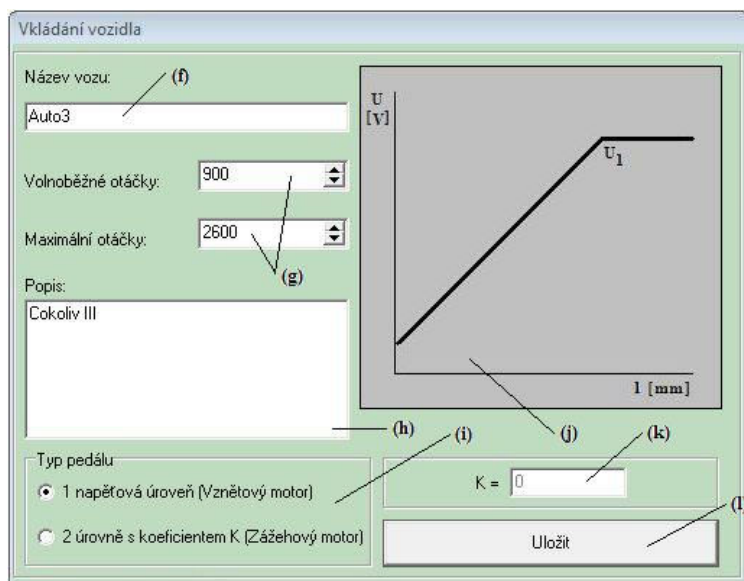
Hlavní okno programu, zobrazené na obr. 32, poskytuje všechny funkce programu v horním roletovém menu (a), které je běžné jako ve většině aplikací pod OS Windows. V hlavní tabulce (b) je zobrazena aktuálně otevřená databáze. Kliknutím na položku v tabulce se položka označí jako aktuálně zvolená a její název se zobrazí mezi parametry měřicí sady. Všechny další parametry měřicí sady se nastavují v oblasti (c) pod tabulkou. Kliknutím pravým tlačítkem do tabulky se zobrazí lokální roletové menu (d). Položky v tomto menu jsou různé podle aktuálního stavu databáze. Není-li databáze načtená, zobrazí se v tomto menu pouze tlačítko pro načtení databáze nebo měřicí databáze. Při načtené databázi jsou v tomto menu tlačítka pro editaci, přidání nebo odebrání položky v databázi (viz obr. 32). Informace o aktuálním stavu programu jako například aktuálně otevřená databáze nebo připojení k portu se zobrazují v dolní stavové liště (e).

Okno pro vkládání a editování automobilu

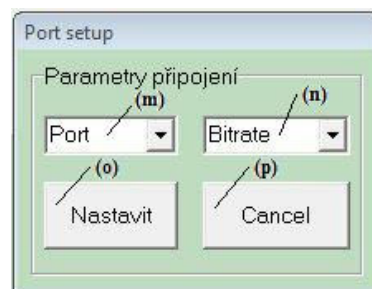
Při přidávání do databáze nebo editování již vytvořeného vozidla se používá druhé okno programu. Toto okno (obr. 33) slouží k nastavení všech potřebných parametrů automobilu. Pro nastavení parametrů slouží různé ovládací prvky formuláře (A-F). Tlačítkem „Uložit“ (G) se data z formuláře uloží do databáze a formulář se zavře.

Okno Port setup

Připojení přístroje vyžaduje nastavení přenosové rychlosti a portu, na kterém je přístroj připojen (H, I). Tyto parametry se nastavují v okně, které je zobrazeno na obr. 34.



Obr. 33: Okno pro vkládání a editování automobilu ovládací aplikace Gas control



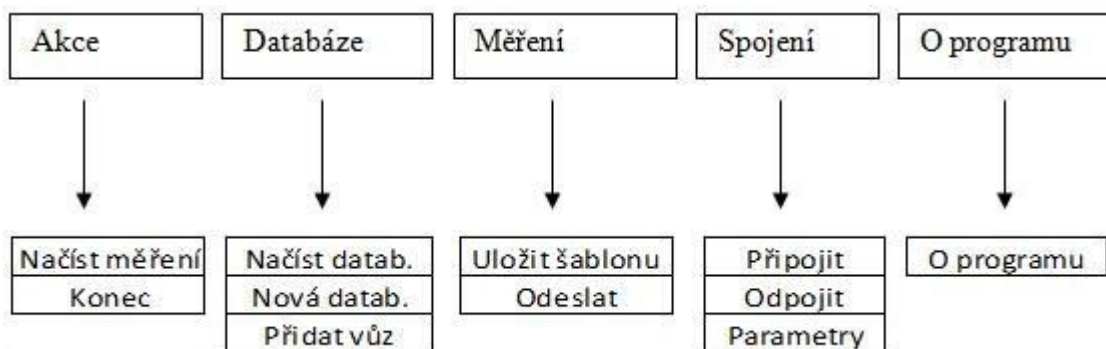
Obr. 34: Okno pro nastavení parametrů připojení

Ovládací prvky formulářů na obr. 33 a 34:

- f** – Pole se jménem vozidla
- g** – Ovládací prvky pro nastavení volnoběžných a maximálních otáček (viz 3.2)
- h** – Pole pro vkládání poznámek (např. detaily o prototypu, motorizace atd.)
- i** – Prvek pro volbu typu plynového pedálu
- j** – Obrázek znázorňující závislost napětí zvoleného typu plynového pedálu na jeho sešlápnutí
- k** – Pole pro vložení koeficientu pro druhé napětí (výchozí hodnota 2 je typickou hodnotou pro všechny dosavadní automobily s tímto druhem pedálu)
- l** – Tlačítko pro uložení vytvořeného automobilu
- m** – Roletové menu pro nastavení čísla portu, na který je přístroj připojen (viz 3.2)
- n** – Roletové menu pro nastavení přenosové rychlosti (pro základní verzi přístroje 115200 kbps)
- o** – Tlačítko pro uložení nastavených parametrů
- p** – Odloží nastavení parametrů na pozdější dobu (ponechá výchozí nastavení COM1 a 115200 kbps)

Horní roletové menu

Všechny funkce programu jsou dostupné přes horní roletové menu. Schéma všech položek menu je zobrazeno na obr. 35. Význam jednotlivých položek bude vysvětlen dále.



Obr. 35: Schéma hlavního roletového menu ovládací aplikace Gas control

3.4 Pokyny a doporučené postupy

3.4.1 Instalace aplikace

Aplikace Gas control je přímo spustitelná pomocí souboru „Gas control.exe“, takže nevyžaduje žádnou instalaci. Pro správnou funkci přístroje je ovšem nutné nainstalovat ovladače FTDI čipu, který je v přístroji použit pro komunikaci přes USB rozhraní. Ovladač VCP (Virtual Com Port) vytvoří v počítači nový port COM. Prostřednictvím tohoto portu se následně aplikace připojuje k přístroji. Postup instalace VCP ovladače:

- 1) Získání ovladačů – Na CD s programem je v adresáři „Ovladače“ přiložen ovladač VCP, který byl v době vzniku tohoto manuálu aktuální pro Windows XP Service Pack 2. Pokud by bylo zapotřebí, aktuální ovladače pro všechny podporované operační systémy, jsou k dispozici na webu výrobce čipu (www.ftdichip.com/Drivers/VCP.htm).
- 2) Samotná instalace – Ještě **před prvním připojením přístroje** k USB počítače spustit instalační soubor získaného VCP ovladače a nainstalovat ho na PC. Pokud jste připojili přístroj a počítač Vás vyzval k instalaci ovladačů, vyberte možnost instalace ovladačů z pevného umístění a vyberte zmíněný ovladač z bodu (1). Pokud je počítač připojený k internetu a po připojení přístroje začal samovolně stahovat a instalovat ovladače, dojde ve většině případů ke špatné instalaci ovladačů. Tyto ovladače je následně potřeba odinstalovat ve správci zařízení, odpojit přístroj a pokračovat od začátku bodu (2).
- 3) Připojení přístroje – Po nainstalování a připojení přístroje by ho měl počítač rozpoznat jako zařízení připojené na COM portu. Tuto skutečnost lze ověřit ve správci zařízení.
- 4) Nastavení parametrů – V nainstalovaném ovladači, který je po instalaci nastavený na volný port COM v počítači je zapotřebí nastavit číslo portu, pod kterým se bude přístroj hlásit. Toto přiřazení není nutné, pokud při instalaci ovladače došlo k nastavení čísla portu v rozmezí COM1 - COM12. Tento rozsah aplikace umožňuje ovládat. Pokud je nastavení mimo tento rozsah, je potřeba ho nastavit na jiný volný port v povoleném rozsahu.

- 5) Dalšími parametry, které je nutné nastavit pro bezchybnou komunikaci je komunikační rychlost (v aktuální verzi firmwaru přístroje nastavená na 115200 kbps), počet datových bitů (8) a počet stopbitů (1). Nastavení ovladače je přístupné přes správce zařízení. V sekci porty COM a LPT je přístroj dostupný, pokud je připojený a ovladač správně nainstalovaný. Při otevření jeho podrobností lze nastavovat výše zmíněné parametry jeho ovladače.
- 6) Aplikace je připravena k použití s přístrojem.

3.4.2 Používání aplikace

- 1) Po spuštění se otevře hlavní okno programu. Žádná databáze není načtená. Pokud již máte vytvořenou databázi, můžete ji otevřít. Okno pro otevření souboru se vyvolá kliknutím pravým tlačítkem na prázdnou tabulku nebo tlačítkem Databáze/Načíst databázi v hlavním menu.
- 2) Při prvním spuštění programu je nutné aplikaci vytvořit. Aplikaci vytvoříte kliknutím na tlačítko Databáze/Nová databáze v hlavním menu. Program vytvoří na zadaném umístění soubor, který bude mít podobu souboru (Zvolené_jméno.CAR).
- 3) Do databáze vložte tlačítkem Databáze/Nový vůz nový automobil. Databázi není potřeba nikdy ukládat. Při každé změně se průběžně ukládá do původního souboru. Pokud již máte založenou databázi body 2 a 3 můžete vynechat⁴.
- 4) Pokud je již v databázi požadovaný automobil označte ho kliknutím. Celý řádek se zvoleným automobilem se označí a pod tabulkou se pro kontrolu nastaví jméno zvoleného automobilu.
- 5) Zadejte všechny ostatní parametry testovací sady, jako maximální čas testu a požadované zrychlení v průběhu testu.

⁴ Databází může být vytvořeno libovolný počet. V aplikaci lze pak kdykoliv načíst jakoukoliv databázi.

- 6) Takto vytvořenou sadu měření můžete uložit tlačítkem Měření/Uložit šablonu nebo pokračovat odesláním do přístroje. Po uložení šablony je na vybraném umístění vytvořen soubor, který bude mít podobu souboru (Zvolené_jméno.MEAS). Tuto šablonu lze v programu načíst místo načtení celé databáze. Při načtení šablony měřicí sady tlačítkem Akce/Načíst šablonu se v tabulce zobrazí jen jediný automobil, který je uložen v šabloně, a zároveň se nastaví všechny parametry testovací sady, takže lze ihned pokračovat odesláním dat do přístroje.
- 7) Před odesláním dat do přístroje nastavte komunikační parametry. Pokud není přístroj připojen, připojte ho. Pokud nevíte, na jakém portu je přístroj připojen, zjistěte to například ve správci zařízení (viz výše). Tlačítkem Spojení/Parametry otevřete okno s nastavením parametrů, kde nastavte zjištěné číslo portu a komunikační rychlost (standardně nastavenou v přístroji na 115200 kbps). V levé části dolní stavové lišty se zobrazí zvolený port.
- 8) Tlačítkem Spojení/Připojit provedete spojení s přístrojem. Pokud je spojení úspěšně navázáno, u zvoleného portu ve stavové liště se objeví „Connected“.
- 9) Tlačítkem Měření/Odeslat odešlete soubor do přístroje. Pokud přístroj přijme správně celou měřicí sadu, zelené diody začnou blikat (viz 3.2.5).
- 10) Případné editace již založeného automobilu nebo jeho vymazání z databáze provedete kliknutím pravým tlačítkem nad označenou položkou v tabulce a tlačítkem Editovat položku nebo Vymazat položku.
- 11) Aplikaci je možné kdykoliv ukončit tlačítkem Akce/Konec nebo křížkem v pravém horním rohu. Databázi není nutné ukládat!

3.4.3 Používání sady Gas control

Při používání sady Gas control je bezpodmínečně nutné mít k dispozici přístroj i počítačovou aplikaci. Používání je jednoduché a intuitivní. Postup při použití je následovný:

- 1) Připojte přístroj k počítači s nainstalovanou aplikací Gas control.

- 2) Vytvořte měřicí sadu a odešlete jí do přístroje (viz 3.4.2).
- 3) Připojte přístroj k automobilu (Připojit konektor pedálu a CAN busu)⁵
- 4) Zapněte zapalování automobilu a počkejte až COTR O3 otevře komunikaci s automobilem (červená dioda na COTR 03 přestane blikat).
- 5) Nastartujte automobil a vyzkoušejte, zda automobil reaguje na plynový pedál v režimu „Gas only“. Pokud ne, zkontrolujte připojení konektorů na pedálu.
- 6) Spusťte automatický režim a nechte proběhnout program.
- 7) Po skončení programu můžete program znovu spustit, nahrát jiný program nebo pokračovat v režimu „Gas only“.

3.4.4 Kalibrace koncového stupně pro pedál

Řídicí jednotka motoru, při použití varianty pedálu se dvěma napěťovými úrovněmi, striktně hlídá souběh napětí na pedálu. Pokud není souběh napětí dodržen, automobil nahlásí chybu EPC nebo chybu motorové řídicí jednotky. Souběh napětí zabezpečuje koncový stupeň. V tranzistorovém zesilovacím stupni, je odporový trimr, kterým lze napěťové zesílení koncového stupně kalibrovat na jeho nominální hodnotu (u všech současných typů hodnota 2). Postup pro kalibraci je následovný:

- 1) Připojte přístroj do automobilu pouze ke sběrnici CAN (na napájecí napětí).
- 2) Připojte k přístroji kabel s koncovým stupněm, který chcete kalibrovat.
- 3) Delším držením tlačítka „Režim“ aktivujte kalibrační režim. Kalibrační režim je indikován blikáním zelených diod viz (3.2.5).
- 4) Změřte voltmetrem napětí na výstupním pinu koncového stupně (Pin č. 6) proti zemi (Pin č. 5).
- 5) Odporovým trimrem v koncovém stupni nastavte na výstupním pinu koncového stupně (Pin č. 4) napětí s požadovaným koeficientem oproti

⁵ Přístroj může při programování zůstat připojený k automobilu, musí však být přepnutý do režimu Gas only. Doporučený postup je přístroj před naprogramováním od automobilu odpojit nebo alespoň vypnout zapalování.

napětí naměřenému v předchozím bodě, jak ukazuje vztah (7). Napětí měřte proti zemi (Pin č. 3). Koeficient $k = 2$ je společný pro všechny současné modely Škoda auto.

$$U_{PIN4} = k \times U_{PIN6} \quad (7)$$

- 6) Ukončete kalibrační režim stisknutím tlačítka „Start/Stop“.

4 Zhodnocení dosažených výsledků

V průběhu testování přístroje byly odhaleny chyby, které ve velké míře vznikly nedostatečným teoretickým rozбором. Tyto nedostatky byly opraveny a pouze potvrzují fakt, že teoretickému rozboru je potřeba věnovat až 60 % celkového času při práci na jakémkoliv projektu.

Přístroj byl kromě drobných změn v hardwaru oproti prvotním záměrům doplněn o koncový stupeň přímo na připojovacím kabelu k plynovému pedálu. Touto změnou byla vyeliminována všechna chybová hlášení, která při počátečních testování výrazně ztěžovala ovládání automobilu. Koncový stupeň navíc umožňuje kalibraci výstupních napěťových úrovní, takže lze zaručit, že po kalibraci bude přístroj pracovat spolehlivě dlouhou dobu.

Přístroj je v současné verzi připojován k diagnostické sběrnici automobilu pomocí stávajících zařízení laboratoře, která umožňují přístroji užívat data v diagnostické zásuvce ve voze, i když data, se kterými přístroj pracuje, v této zásuvce běžně dostupná nejsou. Přístroj by o ně mohl požádat sám a komunikaci tak navázat stejně i bez jiných zařízení, ale pokud by se tato procedura měla implementovat přímo do kódu, trval by vývoj podstatně delší dobu a práce by svým rozsahem výrazně přesahovala rozsah bakalářské práce.

Aplikace i přístroj samotný splňují všechny body zadání, ale pro komfortnější ovládání a komplexnější používání při testech by se daly obě části ještě rozšířit. Do přístroje by uživatel mohl nahrát více programů najednou. Mezi těmito programy by následně vybíral pomocí ovládacích prvků přístroje, které by byly složitější. Přístroj by se musel vybavit displejem. V ovládací aplikaci by mohla být implementována podpora pro generování různých průběhů závislosti rychlosti na čase. Přístroj by pak byl schopný, kromě současné lineární závislosti, generovat například exponenciální závislost nebo skokovou změnu. Tento průběh (tzv. rychlý rozjezd) je při zkouškách vozidel také používán, ale je doposud úspěšně obsluhován řidičem. Pro kvalitnější řízení by se mohly také konstanty pro výpočet nastavovat pro každý automobil zvlášť na základě předchozích měření.

5 Závěr

V průběhu testování přístroje i ovládacího programu se podařilo sadu odladit do podoby, ve které je schopná splnit všechny body zadání. Při získání více času pro testování s automobilem na měřicích válcích by bylo ještě možné přístroj lépe nastavit. Konstanty pro regulační výpočty by se daly vyzkoušet na více typech automobilů a motorizací a tím nastavit přístroj optimálně pro širší spektrum automobilů.

Při testování konečné verze přístroje a ovládacího programu byl přístroj vyzkoušen v řízení při testech, pro které je navržen. Přístroj v těchto testech obstál a dokázal automobil řídit podle požadovaného zadání. Na základě těchto výsledků lze považovat práci za splněnou.

Seznam literatury:

- [1] Silicon laboratories: C8051F040 datasheet, revision 1.4, 2004.
- [2] Robert Bosch GmbH: C_CAN, User's manual, revision 1.2, 2000.
- [3] Robert Bosch GmbH: CAN Specification, version 2.0, 1991.
- [4] CAN in Automation, 2008, URL: <<http://www.can-cia.org>>
- [5] Future Technology Devices International Ltd: FT232RL datasheet, version 1.04, 2005.
- [6] Infineon technologies, CAN-Transceiver TLE 6250 datasheet, version 3.4, 2002.
- [7] ASIX s.r.o., To nejlepší z elektroniky, 2008, URL: <www.asix.cz>
- [8] Infineon technologies, Dual Low-Drop Voltage Regulator TLE 4476 datasheet, revision 2.3, 2002.
- [9] Microsoft, Microsoft Developer Network, 2008, URL: <msdn2.microsoft.com>